VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ

Fakulta elektrotechniky
a komunikačních technologií

BAKALÁŘSKÁ PRÁCE

Brno, 2019 Dominik Galád

VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ
BRNO UNIVERSITY OF TECHNOLOGY

FAKULTA ELEKTROTECHNIKY A KOMUNIKAČNÍCH TECHNOLOGIÍ
FACULTY OF ELECTRICAL ENGINEERING AND COMMUNICATION

ÚSTAV TELEKOMUNIKACÍ
DEPARTMENT OF TELECOMMUNICATIONS

IDENTIFIKACE INKOUSTOVÝCH NÁPLNÍ PRŮMYSLOVÉ TISKÁRNY
IDENTIFICATION OF INK CARTRIDGES OF INDUSTRIAL PRINTER

BAKALÁŘSKÁ PRÁCE
BACHELOR'S THESIS

AUTOR PRÁCE — **Dominik Galád**
AUTHOR

VEDOUCÍ PRÁCE — **doc. Ing. Petr Číka, Ph.D.**
SUPERVISOR

BRNO 2019

VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ
FAKULTA ELEKTROTECHNIKY A KOMUNIKAČNÍCH TECHNOLOGIÍ

Bakalářská práce

bakalářský studijní obor **Teleinformatika**
Ústav telekomunikací

Student: Dominik Galád ***ID:*** 195300
Ročník: 3 ***Akademický rok:*** 2018/19

NÁZEV TÉMATU:

Identifikace inkoustových náplní průmyslové tiskárny

POKYNY PRO VYPRACOVÁNÍ:

Bakalářská práce se bude věnovat identifikaci inkoustových náplní do průmyslových tiskáren TSJet. Během řešení práce student:

1. popíše základní přehled možností identifikace inkoustových náplní,
2. vybere vhodné řešení pro průmyslové inkoustové tiskárny TSjet, použitelné v praktickém nasazení tiskáren,
3. vybere vhodný algoritmus identifikace tiskové náplně s ohledem na zabránění použití neoriginální náplně,
4. navrhne prototypové řešení za účelem ověření jeho základních vlastností,
5. ověří vhodný algoritmus identifikace tiskové náplně s ohledem na zabránění použití neoriginální náplně v prototypovém systému,
6. navrhne způsob praktické realizace systému v praxi,
7. realizuje navržený systém formou prototypu a ověří jeho činnost v reálných podmínkách,
8. na základě praktických poznatků získaných realizací prototypu zhodnotí úroveň bezpečnosti navrženého řešení a posoudí možnost jeho prolomení neautorizovanou osobou,
9. naznačí možnosti použití tohoto řešení pro identifikaci a autorizaci HW a SW tiskáren TSjet s ohledem na servis a update systémů u zákazníka.

DOPORUČENÁ LITERATURA:

[1] GARDNER, Nigel. PICmicro MCU C: An introduction to Programming the Microchip PIC in CCS C. Revised Edition. -: Ccs Inc; Revised edition, 202n. l. ISBN 978-0972418102.

[2] VERLE, Milan. PIC Microcontrollers - Programming in C. -: mikroElektronika, 2009. ISBN 978-8684417178.

Termín zadání: 1.2.2019 ***Termín odevzdání:*** 27.5.2019

Vedoucí práce: doc. Ing. Petr Číka, Ph.D.
Konzultant: Jiří Vrtělka (TS Electronics Zlín, s.r.o.)

prof. Ing. Jiří Mišurec, CSc.
předseda oborové rady

ABSTRAKT

Tato práce se zabývá identifikací inkoustových cartridgí průmyslovou tiskárnou TSjet. Jsou zde představeny základní způsoby identifikace, které se v dnešní době běžně používají. Dále je zde popsána sběrnice 1-Wire® a rozhraní USART. Ve druhé části se práce zabývá vhodnými zabezpečovacími algoritmy pro šifrování dat uložených na identifikačním čipu. V praktické části je popsán návrh systému pro identifikaci inkoustové cartridge. V závěru práce jsou shrnuty výsledky testování návrhu v praxi a hodnocení bezpečnosti celého sytému.

KLÍČOVÁ SLOVA

Identifikace cartridge; tiskárna TSjet; sběrnice 1-Wire®; rozhraní USART; paměť DS2431; DES; AES 128; PIC16F18326

ABSTRACT

This thesis deals with the identification of ink cartridges by the industrial printer TSjet. There are described the basic methods of identification of ink cartridges that are commonly used today. There is also described 1-Wire® bus and USART interface. In the second part the thesis deals with suitable security algorithms for data security stored on the identification chip. The practical part describes the design of the system for identification of the ink cartridge. The conclusion of the thesis summarizes the results of design testing in practice and the safety evaluation of the whole system.

KEYWORDS

Cartridge identification; TSjet printer; 1-Wire® bus; USART interface; memory DS2431; DES; AES 128; PIC16F18326

GALÁD, Dominik. *Identifikace inkoustových náplní průmyslové tiskárny*. Brno, 2018, 58 s. Bakalářská práce. Vysoké učení technické v Brně, Fakulta elektrotechniky a komunikačních technologií, Ústav telekomunikací. Vedoucí práce: doc. Ing. Petr Číka, Ph.D.

PROHLÁŠENÍ

Prohlašuji, že svou bakalářskou práci na téma „Identifikace inkoustových náplní průmyslové tiskárny“ jsem vypracoval samostatně pod vedením vedoucího bakalářské práce a s použitím odborné literatury a dalších informačních zdrojů, které jsou všechny citovány v práci a uvedeny v seznamu literatury na konci práce.

Jako autor uvedené bakalářské práce dále prohlašuji, že v souvislosti s vytvořením této bakalářské práce jsem neporušil autorská práva třetích osob, zejména jsem nezasáhl nedovoleným způsobem do cizích autorských práv osobnostních a/nebo majetkových a jsem si plně vědom následků porušení ustanovení § 11 a následujících autorského zákona č. 121/2000 Sb., o právu autorském, o právech souvisejících s právem autorským a o změně některých zákonů (autorský zákon), ve znění pozdějších předpisů, včetně možných trestněprávních důsledků vyplývajících z ustanovení části druhé, hlavy VI. díl 4 Trestního zákoníku č. 40/2009 Sb.

Brno ....

podpis autora

PODĚKOVÁNÍ

Rád bych poděkoval vedoucímu bakalářské práce panu doc. Ing. Petru Číkovi Ph.D. za odborné vedení, konzultace, trpělivost a podnětné návrhy k práci. Dále bych rád poděkoval firmě TS Electronics Zlín, s. r. o., která se podílela na vedení bakalářské práce, především Ing. Jiřímu Vrtělkovi a Ing. Josefu Janáčovi za trpělivost během konzultací, podmětné návrhy a připomínky k práci.

Brno ....

podpis autora

Tato práce vznikla jako součást klíčové aktivity KA6 - Individuální výuka a zapojení studentů bakalářských a magisterských studijních programů do výzkumu v rámci projektu OP VVV Vytvoření double-degree doktorského studijního programu Elektronika a informační technologie a vytvoření doktorského studijního programu Informační bezpečnost, reg. č. CZ.02.2.69/0.0/0.0/16_018/0002575.

EVROPSKÁ UNIE
Evropské strukturální a investiční fondy
Operační program Výzkum, vývoj a vzdělávání

MINISTERSTVO ŠKOLSTVÍ, MLÁDEŽE A TĚLOVÝCHOVY

Projekt je spolufinancován Evropskou unií.

Obsah

Seznam obrázků

Seznam tabulek

Seznam výpisů

Úvod

Cartridge slouží k uchování inkoustu, který se používá u tiskáren jako médium k převodu digitálních obrazců na papír případně jiný podkladový materiál. Inkoustové tiskárny se používají při různých aplikacích. Nejčastěji se inkoustová tiskárna vyskytuje v domácnostech, protože její pořizovací cena je nízká. Také se běžně nachází v prostředí tiskových center, kde převážně slouží k velkoformátovému tisku, takovým zástupcem jsou plotry. Již méně často bývají používány v kancelářských prostorech, kvůli nižšímu počtu vytisknutých stránek za minutu, který je způsoben technologií tisku. Dále se nasazují v průmyslu, kde se používají převážně k označování výrobků. Zde se volí inkoust s nejvhodnějším chemickým složením podle podkladového materiálu, na který se značení nanáší. Takovým značením může být datum spotřeby a šarže, které se nanášejí na přesně určené místo již předem připraveného obalového materiálu, běžně dodávaného ve velkých rolích. Nebo se aplikuje značení na etikety, které se posléze nalepují na hotový výrobek, takovým zástupcem jsou kupříkladu vinné láhve.

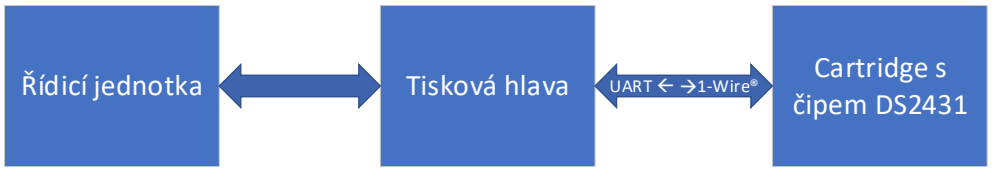
Práce se zabývá problematikou identifikace cartridge u průmyslových tiskáren TSjet od firmy TS Electronics Zlín, s.r.o.. Jsou tvořeny hlavní řídicí jednotkou a tiskovou hlavou, volitelně i senzory sledujícími různé parametry při tisku. Zde ještě výrobce nepoužívá žádnou formu identifikace svých inkoustových cartridgí a nemůže tedy zajistit kvalitu tisku ani bezproblémový chod zařízení. To je hlavním důvodem, proč vznikl požadavek na identifikaci cartridge.

V první části práce jsou popisovány různé možnosti, jak zabránit použití neoriginální cartridge v tiskárně. Je zvolen nejvhodnější způsob tohoto zabezpečení s ohledem na použití v praxi. Druhá část je věnována podrobnějšímu popisu zvolené možnosti ověření originality inkoustové cartridge do tiskárny. Je zde také vybráno konečné řešení jednoznačné identifikace. Ve třetí části práce jsou popisovány vhodné možnosti aplikace algoritmu, který má zabránit použití neoriginální inkoustové náplně, předposlední části je navrženo prototypové řešení s ohledem na další využití v praxi. Závěrečná část popisuje aplikaci šifrovací funkce na mikrokontroléru a návrh desky plošného spoje.

1 Identifikace inkoustové cartridge

Existují jen dvě možnosti jak rozpoznat, že se jedná skutečně o náplň dodanou výrobcem tiskárny. Rozpoznání těchto cartridgí je prováděno buď na základě jejich fyzických rozměrů a tvarů nebo využitím identifikačního čipu:

- **Identifikace na základě fyzických rozměrů a tvaru.** Ve své podstatě je každá cartridge průmyslovým vzorem nebo patentem a vztahují se na ní autorská práva. Chránit cartridge tímto způsobem je už v dnešní době neúčinné a neekonomické. Největším problémem zůstává právě ekonomické hledisko, protože musí být brány v potaz náklady na návrh vlastní cartridge a náklady spojené s její výrobou. Dále je potřeba zajistit inkousty pro tyto cartridge. Obzvláště jedná-li se o výrobu v řádech několika stovek milionů kusů, vždy se najde firma, která začne vyrábět náhrady originálů. Jsou také často levnější, protože výrobce náhrad nemá náklady na jejich vývoj. Neoriginální výrobky však musejí mít stejné parametry jako originál. To lze zajistit pouze zakoupením originální cartridge a okopírováním jejich vlastností Tyto cartridge jsou pak naplněny levným a nekvalitním inkoustem. Proto si zabezpečení tímto způsobem mohou dovolit jen výrobci komerčních tiskáren, které se běžně používají v domácnosti nebo kancelářích, a je předem známo, že budou mít velký odběr spotřebního materiálu, tudíž se jim náklady investované do vývoje se vrátí.
  Četnost použití průmyslových tiskáren je značně menší, a proto by zabezpečení fyzickým tvarem bylo neefektivní a neekonomické.
  Jednou z prvních firem, která začala vyrábět průmyslové tiskárny založené na principu vyměnitelných cartridgí byla firma Hewlett-Packard. Použila svoji cartridge známou pod označením HP 45. Protože firma Hewlett-Packard přišla s tímto způsobem jako první a povrchy, na které se tiskne jsou v průmyslu rozmanité, začali výrobci cartridge HP 45 plnit různými typy inkoustů. Díky široké škále inkoustů se stala cartridge HP 45 od firmy Hewlett-Packard standardem v odvětví průmyslových tiskáren.
- **Identifikace za pomocí čipu.**



Obr. 1.1: Principiální schéma zabezpečení pomocí čipu

Metoda je založena na verifikaci obsahu čipu, umístěného na cartridgi. Ten komunikuje skrze tiskovou hlavu s řídicí jednotkou tiskárny a buď je tisk s car-

tridgí povolen, nebo zakázán, případně se do zařízení uloží záznamy o použití neoriginálního spotřebního materiálu. V dnešní době je metoda využívaná všemi výrobci komerčních tiskáren, a to díky své podstatně vyšší efektivitě, než má zabezpečení jen samotným fyzickým tvarem cartridge. Tento způsob ochrany navíc jak sledovat další parametry inkoustové cartridge, jako jsou například hladina inkoustu, výrobce, datum výroby a podobně. Dále zabrání opětovnému použití znovu naplněné cartridge někým jiným než výrobcem tiskárny. Tuto možnost využívá i původní výrobce inkoustové náplně Hewlett-Packard, který cartridge opatřil i vlastním identifikačním čipem. K němu, ale nejsou známy žádné přesné informace a navíc jej často neoriginální cartridge ani neobsahuji.

V praxi výrobci komerčních tiskáren kombinují obě výše popsaná řešení. Vyrábějí si své vlastní inkoustové cartridge různých rozměrů a tvarů a ještě je opatřují i vlastními identifikačními čipy.

Pokud se jedná o komerčně úspěšný typ náplně, najde se spousta výrobců náhrad, kteří investují do „prolomení“ zabezpečení těchto náplní.

Z důvodu úzkého zaměření průmyslové tiskárny není vysoké riziko toho, že bude někdo investovat prostředky do prolomení zabezpečení čipu. Někteří komerční výrobci tiskáren svoje náplně opatřují EEPROM pamětí DS2431. Mezi mě patří například Samsung, Xerox Corporation, Ricoh Company, Ltd. Paměť komunikuje pomocí sběrnice 1-Wire®, ale mikrokontroléry často nejsou opatřeny tímto komunikačním rozhraním, proto se používá rozraní UART v kombinaci s otevřeným drainem. Také je k ní dostupná technická dokumentace a není s ní složité navázat spojení. [2]

1.1 Cartridge pro průmyslové tiskárny

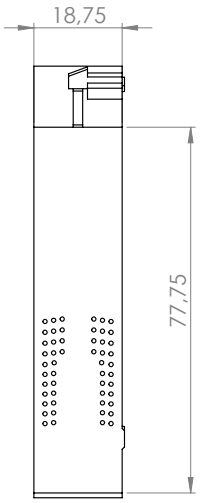
Jak již bylo zmíněno v kapitole 1. , první, kdo začal používat cartridge u průmyslových tiskáren byla společnost Hewlett-Packard a nastavila tím tak standard v odvětví průmyslových tiskáren založených na používání cartridge. Tím se stala cartridge HP 45. Dnes ji již nevyrábí jen společnost Hewlett-Packard, ale i řada jiných výrobců, kteří okopírovali tvar pouzdra cartridge HP 45 a plní ji nejrůznějšími typy inkoustů pro různé podklady a aplikace.

Princip fungování HP45: K uvolnění inkoustu dochází pomocí **termočlánků**. V tomto případě každá tryska obsahuje malý rezistor nebo malý kus kovu. Rezistorem nebo kovem protéká malý proud, v důsledku čehož dojde k zahřátí a odpaření drobné kapky inkoustu, která je otvorem na dně vytlačena ven z cartridge a rozprášena na podklad.[3] Princip uvolňování inkoustu je závislý na trvalé přítomnosti kapaliny, sloužící jako chladivo pro termočlánky,

protože ty jsou citlivé na poškození teplem a nejsou-li chlazeny kapalinou dochází k jejich nevratnému poškození. Tento způsob používají firmy jako Canon, Hewlett-Packard, Lexmark.

Parametry: Původní obsah cartridge je 42 mililitrů inkoustu a jedna kapka obsahuje 33 pikolitrů inkoustu. Inkoustová cartridge má přímo zabudovanou tiskovou hlavu ve svém těle a při každé výměně se s cartridgí vymění i ona. Tisková hlava obsahuje 300 trysek ve dvou řadách, to umožňuje rozlišení až 600 DPI[1]. [4] Díky parametrům se stala oblíbenou v odvětví průmyslových tiskáren a dnes je plněna nejrůznějšími typy inkoustů pro speciální aplikace. Mohou to být například jedlé inkousty, inkousty vytvrzované UV světlem nebo otěru odolné inkousty.

Obr. 1.2: Cartridge HP 45 [24]



Obr. 1.3: Rozměry přední strany cartridge HP 45 [25]

1.2 Řídicí jednotka průmyslové tiskárny

Hlavním stavebním kamenem řídicí jednotky průmyslové tiskárny je zpravidla mikrokontrolér. Jednočipový počítač se označuje anglicky jako microcontroller unit (MCU). Jedná se většinou o jeden monolitický integrovaný obvod, který obsahuje kompletní mikropočítač. Jednočipové počítače se vyznačují velkou spolehlivostí a malými rozměry, proto jsou s oblibou používány v průmyslových aplikacích. Přispívá tomu i fakt, že mají nízkou pořizovací cenu. Integrovaný obvod na sobě nese vše potřebné na to, aby mohl fungovat bez dalších podpůrných obvodů. Obsahuje samotný procesor, paměti pro uložení programu a dat a další podpůrné obvody, jako

[1] Dots per inch - údaj určující kolik obrazových bodů se vejde do délky jednoho palce

jsou například analogově digitální (A/D) převodníky. Rozlišujeme dvě základní architektury mikroprocesorů, a to von Neumannovu a Harvardskou:

- **von Neumannova architektura**
  Pro tuto architekturu je typické, že má společnou paměť pro data i instrukce programu. Uspořádání má výhodu v tom, že není potřeba rozlišovat instrukce pro přístupy k jednotlivým pamětím. Navíc odpadá potřeba mít sběrnici pro každou paměť, stačí pouze jedna společná. Tím se dosáhne jednoduššího návrhu vlastního čipu mikrokontroléru a také je to výhodné při použití externí paměti – snižuje se potřebný počet vstupů a výstupů. Přítomnost jediné sběrnice má i své nevýhody. Největší z nich je rychlost přístupu k pamětím, protože pro instrukce i data se využívá jen jedna sběrnice. Oproti odděleným sběrnicím pro data a instrukce je řešení pomalejší. [5]
- **Harvardská architektura**
  Pro architekturu je typické, že má oddělenou paměť pro data a instrukce. Když jsou sběrnice fyzicky odděleny, mohou být různě široké, proto se běžně vyskytují u 8bitových počítačů sběrnice šířky 12, 14 i 16 bitů. Jednou z největších výhod architektury je rychlost vykonávaní instrukcí, protože data a instrukce mohou být vyčítány téměř zároveň. Nevýhodou je složitější návrh vlastního čipu mikrokontroléru kvůli potřebě většího počtu vstupů a výstupů. [5]

Dále je možné je dělit podle instrukčních sad. Jsou dvě – Reduced Instruction Set Computer (RISC) a Complex Instruction Set Computer (CISC):

- **Reduced Instruction Set Computer**
  Označují se zkratkou RISC. Český překlad zní: počítač pracující s omezenou (zmenšenou) sadou instrukcí. Sada je vysoce optimalizovanou sbírkou strojových instrukcí. Má však nevýhodu v tom, že některé složitější operace musejí být popsány větším počtem příkazů a programový kód ztrácí na svojí hustotě. Naopak jednoduché instrukce v této sadě příkazů trvají vždy právě jeden takt procesoru. Také se přišlo na to, že je velmi složité vytvořit kompilátor, který by dokázal efektivně využívat tyto instrukce. Proto se využívají hlavně tam, kde se programuje na velmi nízké úrovni programovacího jazyka. Například u Assembleru se používají přímo strojové instrukce. [6]
- **Complex Instruction Set Computer**
  Označují se zkratkou CISC z anglického Complex Instruction Set Computer. Překlad do českého jazyka zní: Počítač pracující s kompletní (komplexní) sadou instrukcí. Sada instrukcí obsahuje i složitější funkce jako je například násobení, které se dá nahradit ve strojovém kódu sčítáním a bitovým posunem. Instrukce a doba jejich vykonávání potom mají různou délku. Procesory nemají tak velký počet registrů, ve kterých jsou obsaženy všechny tyto funkce, protože jsou kompilátorem přeloženy na strojově jednodušší, které jsou i fyzicky snadno

realizovatelné, na rozdíl od některých složitějších instrukcí, které se ani nedají fyzicky realizovat. Zdrojový kód pro tyto procesory můžeme psát ve vyšších programovacích jazycích jako je jazyk C. Navíc výsledný počet samotných strojových instrukcí může být menší, protože nepotřebuje totožnou instrukci pro každý registr zvlášť. Postačí nám jedna, ta se vykonává nad všemi registry. Proto jsou dnes tyto procesory hojně rozšířené. [6]

Mezi významné zástupce mikrokontrolérů patří procesory založené na jádře Intel 8051 a mikrokontroléry rodiny Microchip PIC.

- **Intel 8051**

  Procesor postavený na jádře Intel 8051 je používán jako hlavní procesor v řídicí jednotce tiskárny TSjet. Zajišťuje všechny činnosti v řídicí jednotce, od komunikace až po řízení samotné tiskové hlavy.

  Jedná se o 8-bit počítač, který vyvinula firma Intel v roce 1980. Procesor je typu Harvardské architektury a nasazuje se v oblasti vestavěných systémů. V dnešní době je již nahrazován výkonnějšími mikrokontroléry, ale jsou stále svojí instrukční sadou zpětně kompatibilní s původní instrukční sadou tohoto procesoru. Důležitými vlastnostmi této architektury je široká funkčnost. Čip má na sobě vše potřebné pro to, aby mohl fungovat bez dalších podpůrných obvodů. Dále se vyznačuje tím, že je do něj implementováno komunikační rozhraní UART. Dva 16-bit čítače, 8-bit datová sběrnice, několik přerušení, paměti RAM a ROM. [8]

  Jeden ze zástupců mikrokontrolérů, který je postavený na jádře Intel 8051 je i řada **AT89** od firmy Atmel. Z důvodů standardizované instrukční sady, nízké pořizovací ceně a své dostupnosti je tento mikrokontrolér velmi univerzální. V poslední době byl procesor rozšířen o speciální funkce, jako je USB, I^2C, řadiče SPI a CAN sběrnice. Mikrokontrolér se vyrábí v různých variantách. [7]

- **Rodina procesorů Microchip PIC**

  Protože hlavní procesor řídicí jednotky je využíván již na maximum a nebyl by schopen obsloužit další funkce, bylo na základě empirického výzkumu rozhodnuto o použití koprocesoru.

  Jde o mikrokontrolér vyráběný firmou Microchip Technology. Jednočipový počítač je harvardského typu. Mikrokontrolér pracuje s instrukční sadou RISC. Klíčovými vlastnostmi tohoto procesoru jsou oddělená paměť dat a programu, malé množství strojových instrukcí, vykonávání většiny instrukcí v jednom cyklu a není typu load-store (data se nemusejí přesouvat z paměti do registru, aby mohla být zpracována). Jednočipový počítač se vyrábí v mnoha řadách. [9]

  Procesor **PIC 16F18326** je vybaven 28 kilobajty paměti pro program, 2 048 bitovou statickou pamětí (SRAM) pro hodnoty proměnných, které jsou potřeba

pro chod programu. Dále disponuje 256 bitovou pamětí EEPROM (1.5.1), zde data zůstávají uložená i po ztrátě elektrického napájení. Podporuje komunikaci pomocí rozhraní USART (viz 1.3) jedenkrát, dvakrát potom rozhraní SPI a I^2C. Mezi největší výhody mikrokontroléru patří velký rozsah pracovního napětí. Pohybuje se v rozmezí od 1,8 *volt* (V) po 5,5 V. Jednočipový počítač nemá pevně předdefinované piny, jejich funkce se definuje až v programu. To umožňuje přepínat vstupy a výstupy dle potřeby. Další výhodou je, že má integrovaný vnitřní oscilátor s vlastní frekvenci 32 *megahertz* (MHz). To umožňuje využívat vysokých komunikačních rychlostí na jednotlivých rozhraních. [10]

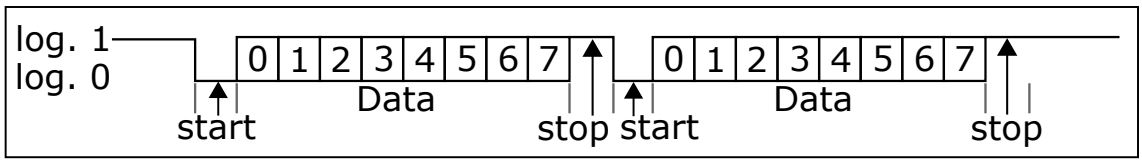
1.3 Univerzální synchronní a asynchronní sériové rozhraní

Jedná se o rozhraní, které slouží k sériové komunikaci a může pracovat v asynchronním režimu nebo v synchronním režimu. Je označováno jako USART. Zkratka vychází z anglického označení Universal Synchronous / Asynchronous Receiver and Transmitter. Rozhraní se používá v mikrokontrolérech od firem Atmel a Microchip, zde v řadě PICmicro®. Je implementováno jako modul a jeho parametry se nastavují dle parametrů zapsaných v jednotlivých registrech, které slouží pro ovládání modulu USART. [9]

Asynchronní přenos

Pro přenos je potřeba nadefinovat dva piny, jeden pro vysílání dat označovaný jako TX z anglického slova transmit a druhý pro příjem dat značený RX z anglického receive. Pokud se nic nevysílá, odpovídá to klidové úrovni v hodnotě logická 1 a přenos může být zahájen kdykoliv. Každý přenášený bajt je označen takzvaným start-bitem, po něm se posílá sled 8 bitů v pořadí od nejnižší významnosti po nejvyšší a následuje stop-bit, ten ukončuje vysílání jednoho bajtu. To je znázorněno na obrázku 1.4 [11]

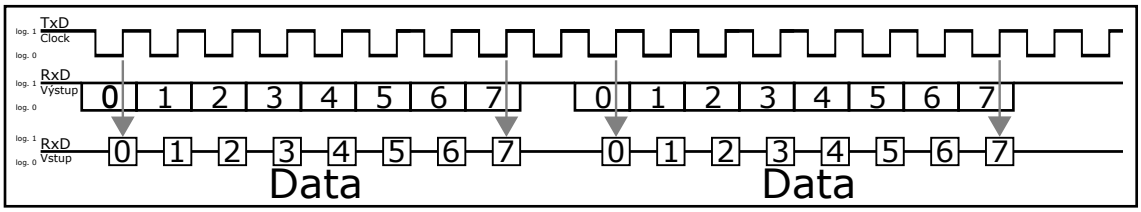
V režimu je zaručena plně duplexní komunikace. To znamená, že data mohou být v jeden moment vysílána i přijímána. [11]



Obr. 1.4: Průběh asynchronního přenosu [26]

Synchronní přenos

Přenos využívá při přenosu dva piny. Jeden slouží jako TX a RX a druhý se využívá pro vysílaní hodinového signálu, který řídí tento přenos. Protože je přenos jen poloduplexní, umožňuje data pouze vysílat nebo přijímat. Nemohou být vysílána a přijímána zároveň. V režimu mohou být zařízení nastavena jako řídicí (Master) a podřízený (Slave) a přenos dat musí být zahájen v daný moment signálem clock. Zachycuje to obrázek 1.5, na něm je pro hodinový signál použit fyzický pin TX a na fyzickém pinu RX se nyní přijímá i vysílá. [11]



Obr. 1.5: Průběh synchronního přenosu dat [27]

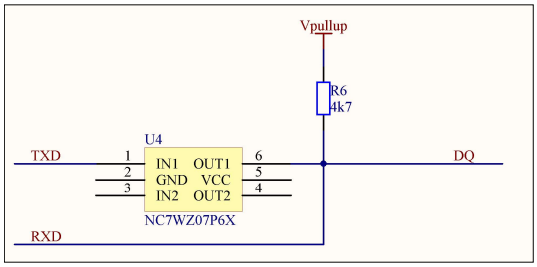
Pro oba režimy je potřeba nastavit *modulační rychlost – Baud rate* (BR). Udává počet symbolů přenesených za sekundu, jinak řečeno, jak rychle se mění stavy přenosového kanálu.

1.4 Sběrnice 1-Wire®

Sběrnice je vyvinutá firmou Dallas Semiconductor Corp. Přenosová rychlost na sběrnici je velmi nízká. Jejím specifikem je, že se zde používají pouze dva vodiče. Jeden slouží jako zem a druhý plní roli napájecího a zároveň i datového vodiče. Proto jsou zařízení komunikující po sběrnici vybavena kondenzátorem o velikosti 800 pikofaradů. V něm se uchovává energie potřebná během komunikace. Sběrnici používají levná a malá zařízení, u kterých se nepřenášejí velké objemy dat, například teploměry, nebo identifikační čipy. Komunikaci vždy řídí hlavní zařízení označované jako master. Zařízení je zpravidla mikrokontrolér, případně počítač. Další zařízení (nazývaná slave) jsou mu podřízena a poslouchají příkazy. Sběrnici může sdílet několik zařízení najednou, každé připojené zařízení má své 64bitové sériové číslo, které je unikátní a definované přímo výrobcem. Prvních 8 bitů z něj označuje typ zařízení, například pro EEPROM paměť je to v hexadecimálním tvaru 2d. Standardní přenosová rychlost je 16,3 *kilobit za sekundu* (kb/s). [13]

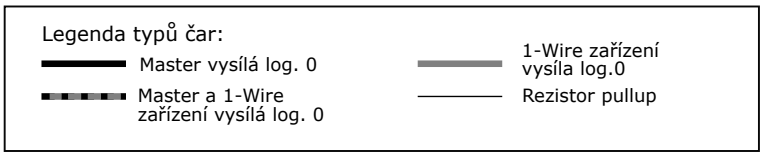
Master zahajuje komunikaci posláním reset-pulsu. To znamená, že sběrnici uzemní na dobu nejméně 480 *mikrosekund* (μs), poté se ohlásí zařízení slave tak, že jakmile je sběrnice uvolněna, posílá logickou (log.) 0 na dobu 60 μs. Pokud chce master poslat log. 1 uzemní sběrnici na dobu 1–15 μs. Pokud chce vyslat log. 0 uzemní sběrnici

na 60 μs. Slave po zaznamenání sestupné hrany naslouchá na sběrnici po dobu 30 μs, proto je tak výrazný časový rozdíl mezi dobou uzemnění pro log. 1 a log. 0. Pokud chce master přijímat, uzemní sběrnici na 1-15 μs. Pokud slave posílá log. 1, pullup rezistor vrátí stav sběrnice do log. 1. Když se vysílá log. 0, je sběrnice stažena k log. 0 na dobu 60 μs. Komunikace standardně probíhá v následujícím sledu. Je poslán reset-puls, následuje 8 bitů příkazu a pak se posílají anebo přijímají data. [13]



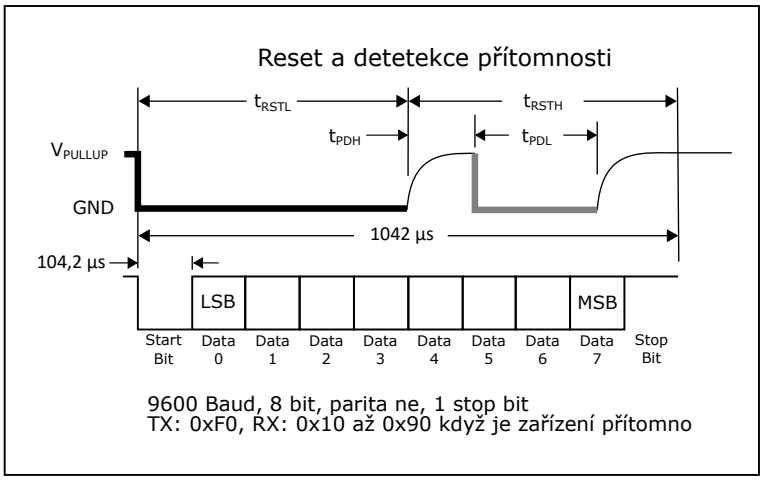
Obr. 1.6: Schéma připojení 1-Wire® zařízení pomocí UART [14]

Pokud zařízení není vybaveno rozhraním pro komunikaci pomocí 1-Wire® sběrnice, může být použito rozhraní UART, kdy TX je opatřen otevřeným kolektorem a RX je připojen za tento kolektor. Dále se ještě použije vhodná velikost pullup rezistoru. Konkrétní zapojení je vyobrazeno na schématu (Obr. 1.6). Zde se jedná o otevřený drain, protože jsou použity tranzistory typu FET, konkrétně typ NC7WZ07. Pin TX je označeno jako TXD a RX obdobně RXD. 1-Wire® zařízení je popsáno jako DQ.



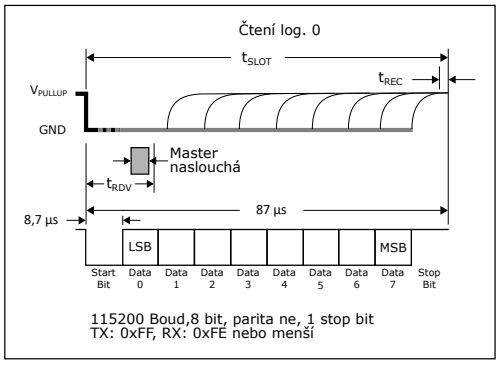
Obr. 1.7: Legenda pro časové průběhy komunikace [14]

Pokud používáme rozhraní UART, je nastaveno tak, aby vysílalo jeden start bit, 8 bitů slova a 1 stop bit.

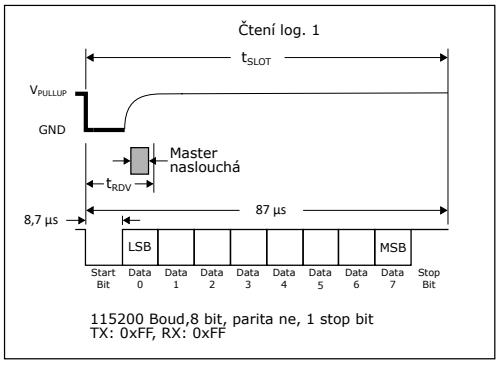


Obr. 1.8: Průběh reset-pulsu [14]

Reset-puls (Obr. 1.8) se posílá v následující formě TX: 0xF0 a vrací se odpověď RX: 0x10 až 0x90 při baut rate 9600. Pokud je požito jen jedno zařízení slave, může být odpověď změněna na 0xE0.

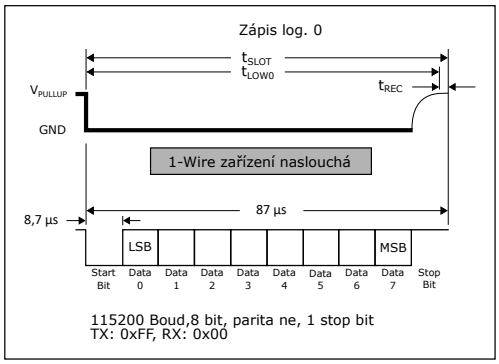


Obr. 1.9: Čtení z 1-Wire® zařízení log. 0 [14]

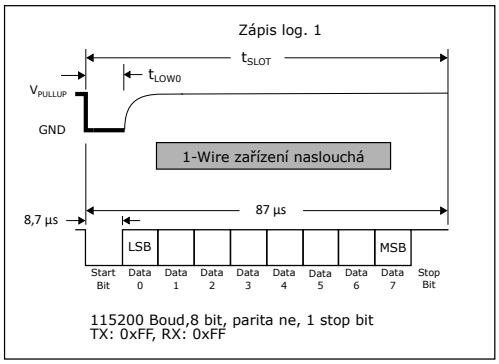


Obr. 1.10: Čtení z 1-Wire® zařízení log. 1 [14]

Při čtení ze zařízení je komunikace následující TX: 0xFF a RX: 0xFE nebo nižší pro log. 0 (Obr. 1.9), pro log. 1 je RX: 0xFF (Obr. 1.10) při baut rate 115200.
Pro zápis log. 1 je TX: 0xFF a RX: 0xFF (Obr. 1.12), u log. 0 je TX: 0x00 a RX: 0x00 (Obr. 1.11) při baut ratu 115200. Výše popsaný postup popisuje čtení anebo zápis jednoho bitu. Pro to, aby byl zapsán nebo přečten jeden bajt, se musí celý postup kromě reset-pulsu opakovat ještě osmkrát. [14]



Obr. 1.11: Vysílání do 1-Wire® zařízení log. 0 [14]



Obr. 1.12: Vysílání do 1-Wire® zařízení log. 1 [14]

1.5 Paměťové médium

Je také označováno jako datové médium nebo záznamové médium. Slouží k uchovávání informací nebo dat. Ukládání je založeno na fyzikálním principu. Za datový nosič lze označit i hmotné nosiče, které uchovávají informace. Nemusí se vždy jednat jen o elektronické datové médium. Paměťovým médiem je v praxi často myšlen jen elektronický datový nosič. Slouží především k uchovávání informací a v případě, že se jedná o externí záznamové médium, také ke sdílení dat mezi jednotlivými zařízeními, které toto médium podporují a+umějí z něj informace vyčíst nebo je na něj zapsat. Nosičem informace je digitální nebo analogový signál. V případě analogového signálu musíme vhodně volit modulaci digitálních veličin. V případě digitálního záznamu se nejčastěji volí binární forma zápisu. Záznam na médiu může mít několik forem:

- Trvalý, nelze jej přepsat.
- Semipermanentní, může být kdykoliv nahrazen novým záznamem.
- Nestálý, například po odpojení napájení dojde k jeho vymazání nebo jej musíme obnovovat v určitém intervalu.

1.5.1 EEPROM Paměť

Jak bylo zmíněno v kapitole 1, výrobci tiskáren opatřují cartridge pamětí DS2431, která je založena na principu EEPROM paměti.

Označení paměti pohází z anglického Electrically Erasable Programmable Read-Only Memory (EEPROM). Paměť je elektronicky mazatelná. Je typu ROM-RAM, tedy buď je určena pouze pro čtení a je jen jednou naprogramována při výrobě, v takovém případě je na ní často uložen firmware, nebo může být používaná jako paměťové medium pro ukládání dat, ta mohou být vyčítána nebo zapisována dle potřeby. Pro paměti je typická životnost okolo 200 000 zápisů, má tedy větší životnost

než paměť typu flash. Nespornou výhodou paměti je i vysoká životnost uložených dat. Běžně i 20 let, v dnešní době však není problém dosáhnout i delší doby. [17] Nevýhodou je ve větší složitost paměťové buňky, což znamená vyšší pořizovací cenu.

Technologie

Pro výrobu EEPROM paměti se používají tranzistory vytvářené technologií Metal Nitrid Oxide Semiconductor (MNOS). Na řídicí elektrodě tranzistoru je nanesena vrstva nitridu křemičitého (Si_3N_4), pod ni se umístí tenká vrstva oxidu křemičitého (SiO_2). [16]

Princip fungování

Paměťová buňka pracuje na jevu zvaném tunelovaní elektrického náboje . Pokud se zapisuje do buňky, je na adresový vodič přivedeno záporné napětí (-U) a datový vodič buňky se uzemní, tím je zapsána hodnota log. 1. Tranzistor se otevře a vznikne na něm náboj, který vytvoří velké prahové napětí. Pokud je paměť vyčítána, je na adresový vodič přiveden záporný impuls. Tranzistor s nízkým prahovým napětím se otevře a vede na datový vodič proud. Naopak tranzistor s velkým prahovým napětím zůstane zavřený. Pro vymazáni paměti je přivedeno kladné napětí na adresový vodič, což způsobí pokles prahového napětí a paměť je vymazána. [16]

1.6 Unipolární tranzistor

Existují dva druhy tranzistorů, bipolární a unipolární. V mikroprocesorové technice se používají převážně unipolární tranzistory kvůli svým vlastnostem, které jsou popsány dále.

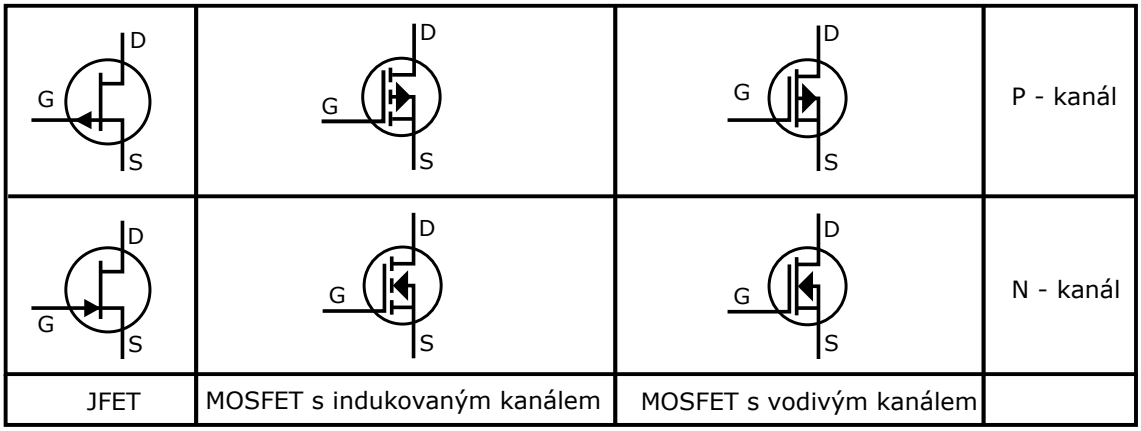
Jedná se o polovodičový prvek. Používá se jako zesilovač, spínač signálů nebo při realizaci logických funkcí. O přenos náboje se starají jen majoritní (většinové) nosiče náboje, proto označení unipolární. Minoritní (menšinové) nosiče náboje jsou v tomto případě parazitní (nežádoucí). Tranzistor je složen ze struktur P a N, jedna je vždy výrazně větší a označuje se jako substrát, na němž je vytvořen zbytek struktury tranzistoru.

Hlavní vlastností je vysoký vstupní odpor, který vyplývá ze struktury tranzistoru. Proto jsou označovány jako tranzistory řízené elektrickým polem, anglicky Field-Effect Transistors (FET). Oproti bipolárnímu tranzistoru, který má malý vstupní odpor, teče vstupní částí unipolárního tranzistoru jen malý nebo takřka žádný proud. To je výhodné pro zdroje s velkým vnitřním odporem, kterými jsou například mikrokontroléry. Dále se v prvním kvadrantu volt-ampérové charakteristiky chovají téměř lineárně, to je výhodou, hlavně když je využíváme jako zesilovač.

Díky jejich fyzické konstrukci se dají realizovat i na malé ploše, což umožňuje to vysokou míru integrace. Nízké pro vedení dovoluje velmi efektivní přenos tepla z přechodu.

Dříve byla nevýhodou u unipolárních tranzistorů velká náchylnost na poškození statickým nábojem kvůli velkému vstupnímu odporu. Manipulace s nimi musela být velmi opatrná a bylo zapotřebí používat speciální oblečení, které nevytváří statický náboj, a uzemňující náramky, aby se předešlo jejich poškození. Dnes jsou již tyto tranzistory standardně vyráběny v pouzdrech, která přímo obsahují ochranné diody. Díky nim odolají bez problému náboji o velikosti jednotek kilovolt.

Unipolární tranzistory se dále dělí podle technologie výroby, té odpovídá i schématická značka (Obr. 1.13).



Obr. 1.13: Schematické značky FET tranzistorů [30]

1.6.1 Otevřený drain

Jedná se o konkrétní zapojení unipolárního tranzistoru. Využívá se v případech, kdy je potřeba zajistit větší proudový odběr zátěže případně sběrnice, například u sběrnice 1-WIRE (1.4), kde jsou zařízení napájena přímo z datové linky. [20]

Elektroda Gate je připojena ve většině případů k výstupu mikrokontroléru, řídí logickou úroveň. Pin s označením Source je připojena na „zem“. K elektrodě Drain se připojuje sběrnice, na které vysíláme. Ta musí být ještě opatřena pullup rezistorem, který zajišťuje logickou úroveň 1 a napájení sběrnice, když je tranzistor zavřený. Přivedeme-li napětí na elektrodu Gate, dojde ke stažení úrovně napění k zemi, tedy log. 0. Zda bude při přivedení napění na vstup tranzistoru na výstupu log. 1 nebo log. 0, záleží na použitém druhu tranzistoru. [20]

2 Zabezpečovací algoritmy dat uložených na identifikačním čipu

Jejich úkolem je zabránit útočníkovi přečíst si obsah zprávy, i když samotná přenášená data může vidět. Jsou založené na matematických operacích v různých soustavách. Zabezpečovací algoritmy nejsou schopny samy zajistit 100% bezpečnost. Spoléhají také na důvěryhodnost systému a organizace, kde jsou používány. Rozlišujeme dva druhy šifry:

Symetrická šifra, šifrovací a dešifrovací klíč jsou si rovny. Šifra je velmi rychlá, ale její nevýhodou je, že obě strany si musí vyměnit klíč. Krok výměny vnáší do celého šifrovacího postupu nespolehlivost, protože existuje velké riziko jeho prozrazení během přenosu. Dále je potřeba mít pro každého účastníka vlastní jedinečný klíč, jinak hrozí prolomení zabezpečení při odposlechnutí více zpráv. [21]

Nesymetrická šifra, zde si nejsou šifrovací a dešifrovací klíč rovny. Šifry jsou oproti symetrickým pomalejší a složitější na výpočet. Používají se zde dva klíče, jeden je veřejný a druhý je soukromý. Dále se pro systém stanoví další parametry. Není zde riziko odhalení klíče při jeho předávání. V tajnosti musí zůstat jen soukromý klíč. [21]

V tomto případě má zabezpečovací algoritmus za úkol zajistit autentičnost a tím ověřit původ inkoustové náplně. Symetrická šifra je vhodnější pro použití na mikrokontroléru, protože je rychlá a není nároční na výpočetní výkon, na rozdíl od nesymetrické šifry, která klada na vysoké nároky na procesor.

Přehled používaných principů symetrických šifer:

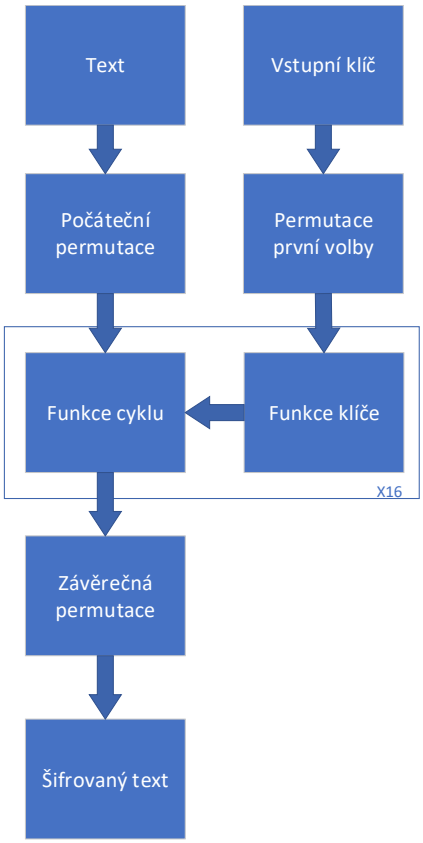
Feistelovo schéma je pojmenováno po Horstu Feistelovi. Výhodou je rychlost a jednoduchost. Základním rysem je rozdělení vstupního bloku na dvě poloviny a ty se poté promíchávají s klíčem. Hlavními zástupci tohoto schématu jsou například Lucifer, Des, 3Des. [22]

Substitučně–permutační síť používá modernější přístup oproti **Feistelovu schématu**. Opakovaně se na blok aplikuje substituce (nahrazování) a permutace (změna pořadí). Nejvýznamnější zástupce je AES. [22]

2.1 Data Encryption Standard (DES)

Jedná se o úspěšný algoritmus, který byl schválen jako šifrovací standard. V dnešní době je ale už zastaralý. Byl prolomen během jednoho dne. Používají se bloky dat o velikosti 64 bit a klíč 56 bit. Na návrhu se podíleli i IBM a NSA. Proti svému původnímu návrhu byla výsledná verze uměle oslabena. Výstupem je kryptogram o délce 64 bit. Průběh šifrování je znázorněn na obrázku 2.1. [22]

Protože byl DES oblíbený v bankovním sektoru, který investoval do drahých šifrovacích zařízení, vznikl 3DES, tedy 3 × DES po sobě. V něm byly odstraněny slabiny původního DESu. Největší slabinou byla malá délka klíče, jen 56 bit. Změnila se na 112 bit a dodnes se používá. 3DES má největší nevýhodu ve své rychlosti, kdy po třech opakováních je už příliš pomalý. [22]



Obr. 2.1: Obecné schéma DES [23]

Dále kapitola vychází z [23].

Popis funkce: Do algoritmu vstupují bloky po 64 bitech. Dojde k počáteční permutaci. Čísla v tabulce určují pořadí bitů ve vstupním bloku. Jsou umístěny na odpovídající místo v matici (Tab. 2.1).

Tab. 2.1: Matice počáteční permutace [23]

58	50	42	34	26	18	10	2
60	52	44	36	28	20	12	4
62	54	46	38	30	22	14	6
64	56	48	40	32	24	16	8
57	49	41	33	25	17	9	1
59	51	43	35	27	19	11	3
61	53	45	37	29	21	13	5
63	55	47	39	31	23	15	7

Do systému (Obr. 2.2) potom vstupuje výstup z matice 2.1, kdy výstupem je 8 bloků po 8 bitech, kde každému bloku odpovídá jeden řádek matice. Výsledkem kroku je blok 64 bitů, který je rozdělen na polovinu. První polovina zůstává nezměněna. Druhá polovina se zvětší pomocí **rozšiřující funkce**. Představuje ji rozšiřující permutační blok, který je realizován maticí 8×6 (Tab. 2.2).

Tab. 2.2: Matice pro rozšíření z 32 bit na 48 bit [23]

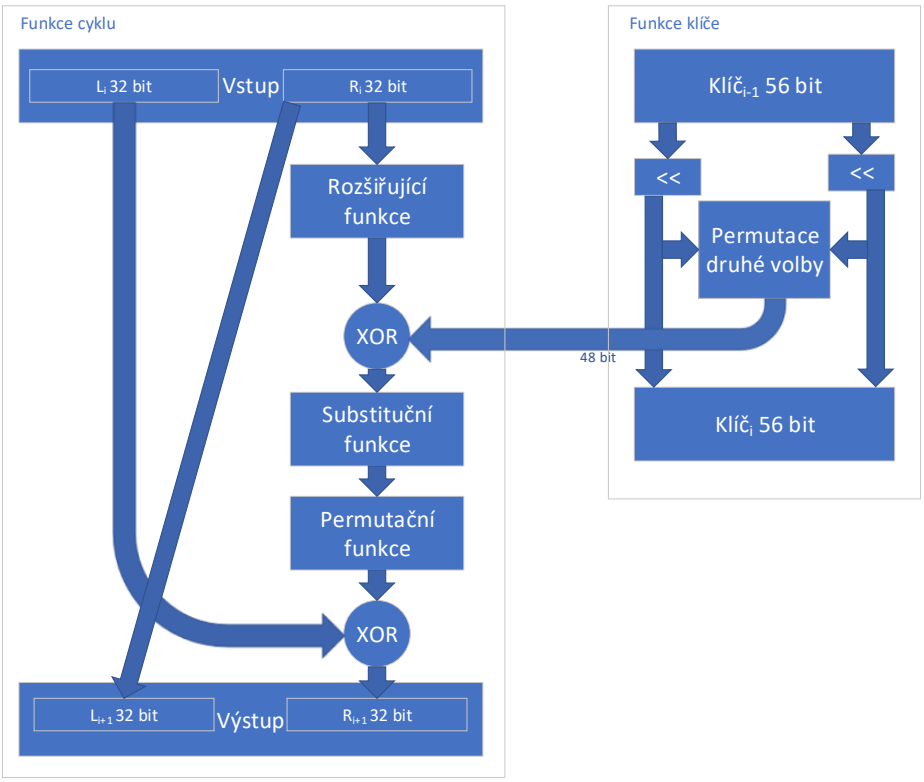
32	1	2	3	5
4	5	6	7	9
8	9	10	11	13
12	13	14	15	17
16	17	18	19	21
20	21	22	23	25
24	25	24	27	29
28	29	30	31	1

Promícháním vznikne z 32 bitů 48 bitů. Krok je nutný k tomu, aby se mohl výstup sečíst s klíčem pro daný cyklus, který má také délku 48 bitů, za pomoci logické funkce XOR. Výstupních 48 bitů se rozdělí na 8 skupin po 6 bitech. Každá skupina se převede pomocí substituční tabulky (Tab. 2.3) na výstup 4 bitů. Hodnoty v tabulce jsou zapsány dekadicky.

Tab. 2.3: Příklad substituční tabulky pro S-Box [23]

	X0000X	X0001X	X0010X	X0011X	X0100X	X0101X	X0110X	X0111X	X1000X	X1001X	X1010X	X1011X	X1100X	X1101X	X1110X	X1111X
0YYYY0	14	4	13	1	2	15	11	8	3	10	6	12	5	9	0	7
0YYYY1	0	15	7	4	14	2	13	1	10	6	12	11	9	5	3	8
1YYYY0	4	1	14	8	13	6	2	11	15	12	9	7	3	10	5	0
1YYYY1	15	12	8	2	4	9	1	7	5	11	3	14	10	0	6	13

Pro každý cyklus existuje vlastní substituční tabulka. Výsledkem **substituční funkce** je slovo o délce 32 bitů. Následuje **permutační funkce**, která je realizována maticí 4×8.



Obr. 2.2: Schéma jednoho cyklu algoritmu DES [23]

Tab. 2.4: Permutační matice [23]

16	7	20	21	29	12	28	17
1	15	23	26	5	18	31	10
2	8	24	14	32	27	3	9
19	13	30	6	22	11	4	25

Čísla zastupují pozici bitu ve výstupu ze substituční funkce. Matice (Tab. 2.4) se vyčítá po řádcích, výsledek je 32 bitů dlouhé číslo. Na závěr se pomocí logické funkce XOR sečte výsledek z permutační funkce s první polovinou 64 bitového vstupního slova. Výsledek je 32 bitů dlouhé číslo. Výsledek z předchozího kroku vstupuje na místo R1 a na místo L1 je dosazený blok R0, který byl vstupem pro předchozí kroky. Celý cyklus se od rozšiřující funkce opakuje ještě 16krát. Po opakovaní se provádí **závěrečná permutace** pomocí matice 8×8 (Tab. 2.5).

Čísla v matici korespondují s pořadím bitů ve výstupním slovu. Šifrované slovo potom odpovídá vyčtení matice po řádcích.

Generace klíčů pro jednotlivé cykly

Vstupní klíč má délku 56 bitů. Před samotným výpočtem klíče pro každý cyklus se provádí **funkce permutace první volby**, ta je definovaná maticí 4×14 (Tab.

Tab. 2.5: Závěrečná permutační matice [23]

40	8	48	16	56	24	64	32
39	7	47	15	55	23	63	31
38	6	46	14	54	22	62	30
37	5	45	13	53	21	61	29
36	4	44	12	52	20	60	28
35	3	43	11	51	19	59	27
34	2	42	10	50	18	58	26
33	1	41	9	49	17	57	25

2.6). Technicky do funkce vstupuje klíč o délce 64 bitů, protože každý 8. bit z bloku je paritní bit. Permutace první volby také zajistí prodloužení klíče. Výstupní klíč je získán vyčtením matice po řádcích.

Tab. 2.6: Matice permutace první volby [23]

57	49	41	33	25	17	9	63	55	47	39	31	23	15
1	58	50	42	34	16	18	7	62	54	46	38	30	22
10	2	59	51	43	35	27	14	6	61	53	45	37	29
19	11	3	60	52	44	36	21	13	5	28	20	12	4

Dalším krokem je bitová rotace doleva. Klíč o délce 56 bitů je rozdělen na polovinu a každá polovina se bitově rotuje o přesně daný počet bitů pro každý cyklus. Počet míst, o kolik se posune bit, je popsán v následující tabulce 2.7:

Tab. 2.7: Tabulka určující počet míst posunu bitu pro každý cyklus [23]

Cyklus	Posun
1	1
2	1
3	2
4	2
5	2
6	2
7	2
8	2
9	1
10	2
11	2
12	2
13	2
14	2
15	2
16	1

Klíč pro jednotlivé cykly získáme z **funkce permutace druhé volby**. Je definovaná maticí 8×6 (Tab. 2.8).

Tab. 2.8: Matice permutace druhé volby [23]

14	17	11	24	1	5
3	28	15	6	21	10
23	19	12	4	26	8
16	7	27	20	13	2
41	52	31	37	47	55
30	40	51	45	33	48
44	49	39	56	34	53
46	42	50	36	29	32

Pro další cyklus je klíč získán z předchozího 56 bitového klíče. Podle tabulky 2.8 se vygeneruje 48 bitů dlouhý klíč pro aktuální cyklus. [23]

Pro dešifrování provedeme celý cyklus jako šifrování, ale klíče do něj vstupují v opačném pořadí. Tedy do prvního cyklu vstupuje 16. klíč. [23]

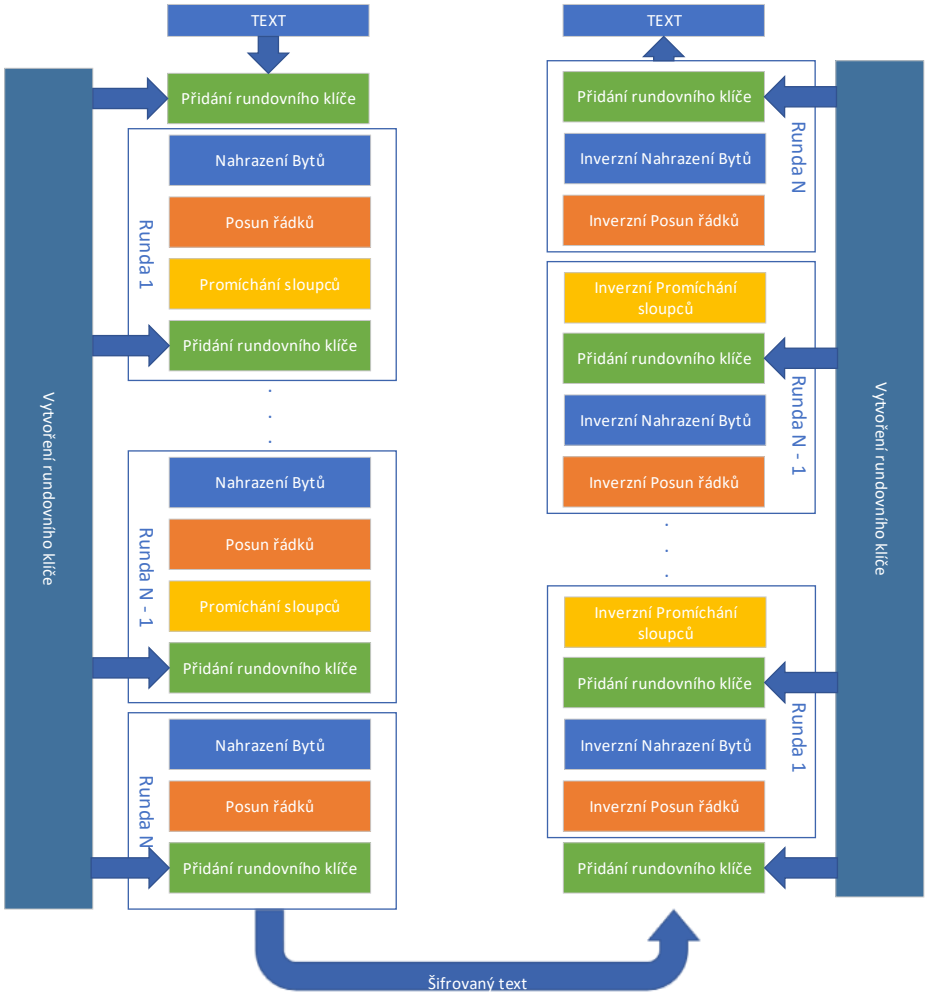
2.2 Advanced Encryption Standard (AES)

Je náhradou za DES. V dnešní době je to standard pro symetrické šifrování. Vytvořili jej Joan Daemen a Vincent Rijmen. Původní název byl Rijndael. K šifrování se používají klíče o délce 128, 192 nebo 256 bitů a bloky o délce 128 bitů. V původním návrhu byly oba parametry volitelné. Pro nejvyšší bezpečnost se doporučují klíče 192 nebo 258 bitů, ale i klíč o délce 128 bitů je nemožné prolomit jen hrubou silou. V dnešní době je považován za bezpečný a velmi rychlý. [22]

Dále kapitola vychází z [31].

Šifrovací proces AES můžeme rozdělit do tří hlavních fází (Obr. 2.3):

1. Inicializační runda
    - Přídání rundovního klíče
2. Hlavní runda
    - Nahrazení Bytů
    - Posun řádků
    - Promíchá sloupců
    - Přidání rundovního klíče
3. Finální runda
    - Nahrazení Bytů
    - Posun řádků
    - Přidání rundovního klíče



Obr. 2.3: Princip fungování algoritmu AES [31]

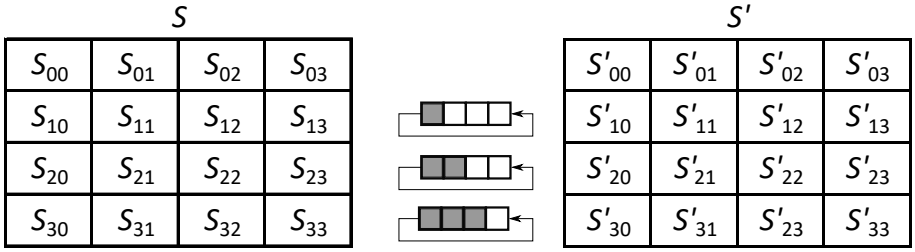
Funkce **přidání rundovního klíče** nemá za úkol nic jiného než sečíst pomocí logické funkce XOR vstupující data s klíčem.

V bloku **nahrazení Bytů** dochází pomocí nahrazovací tabulky k výměně původního Bytu za nový. Ten je dán tabulkou 2.9. Jednotlivé Byty jsou zadány v hexadecimálním tvaru XY. Tedy stačí najít odpovídající sloupec a řádek podle původního Bytu a v místě, kde se protnou je nová hodnota Bytu, kterou nahradíme původní.

Tab. 2.9: Substituční tabulka Bytu XY v hexadecimálním tvaru [32]

		Y															
		0	1	2	3	4	5	6	7	8	9	a	b	c	d	e	f
X	0	63	7c	77	7b	f2	6b	6f	c5	30	01	67	2b	fe	d7	ab	76
	1	ca	82	c9	7d	fa	59	47	f0	ad	d4	a2	af	9c	a4	72	c0
	2	b7	fd	93	26	36	3f	f7	cc	34	a5	e5	f1	71	d8	31	15
	3	04	c7	23	c3	18	96	05	9a	07	12	80	e2	eb	27	b2	75
	4	09	83	2c	1a	1b	6e	5a	a0	52	3b	d6	b3	29	e3	2f	84
	5	53	d1	00	ed	20	fc	b1	5b	6a	cb	be	39	4a	4c	58	cf
	6	d0	ef	aa	fb	43	4d	33	85	45	f9	02	7f	50	3c	9f	a8
	7	51	a3	40	8f	92	9d	38	f5	bc	b6	da	21	10	ff	f3	d2
	8	cd	0c	13	ec	2f	97	44	17	c4	a7	7e	3d	64	5d	19	73
	9	60	81	4f	dc	22	2a	90	88	46	ee	b8	14	de	5e	0b	db
	a	e0	32	3a	0a	49	06	24	5c	c2	d3	ac	62	91	95	e4	79
	b	e7	c8	37	6d	8b	d5	4e	a9	6c	56	f4	ea	65	7a	ae	08
	c	ba	78	25	2e	1c	a6	b4	c6	e8	dd	74	1f	4b	bd	8b	8a
	d	70	3e	b5	66	48	03	f6	0e	61	35	57	b9	86	c1	1d	9e
	e	e1	f8	98	11	69	d9	8e	94	9b	1e	87	e9	ce	55	28	df
	f	8c	a1	89	0d	bf	e6	42	68	41	99	2d	0f	b0	54	bb	16

Dalším krokem je **posun řádků**. Reprezentuje ho matice $\boldsymbol{S}$ 4×4 (Obr. 2.4), tedy její celková velikost je 16 Bytů, což se po vynásobení 8 rovná 128 bitů. Jednotlivé Byty se do matice zapisují po řádcích. Výsledkem kroku je opět matice $\boldsymbol{S'}$ 4×4 (Obr. 2.4). Jen jednotlivé Byty na řádku jsou kruhově posunuty o přesně daný počet míst pro každý řádek.



Obr. 2.4: Ukázka posunutí řádků v matici [32]

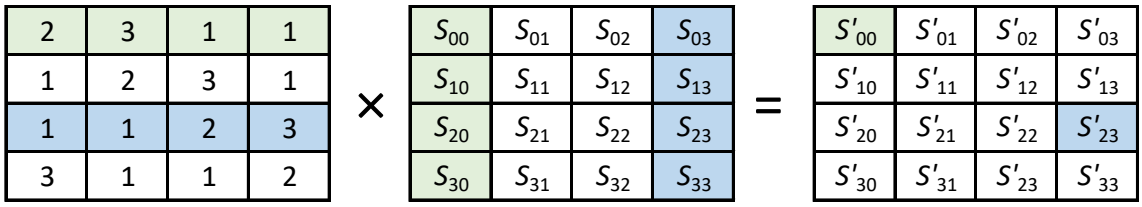
Poté je nutné **promíchat sloupce** a to se provádí pomocí násobení dvou matic.

Z obrázku 2.5 je patrné, že první matice udává řádek a druhá matice udává sloupec, do kterého má být umístěn výsledek. Výsledek je dán vzorcem

$$S'_{00} = 2 \cdot S_{00} \oplus 3 \cdot S_{10} \oplus 1 \cdot S_{20} \oplus 1 \cdot S_{30} \tag{2.1}$$

Pro násobení ve vzorci 2.1 platí, že výsledek musí odpovídat Galois Field[1] 2^8

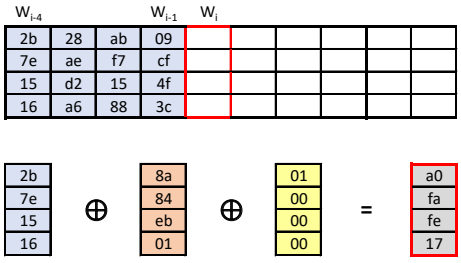
[1]Protože v kryptografii nemůžeme pracovat v neomezeném prostoru, Galois Field = Final Field obě tyto slovní spojení znamenají, že se pohybujeme v konečném poli hodnot. V kryptografii je to standartně 0 až 255. Tedy velikost jednoho bytu, ten je reprezentován 8 bity. Tento prostor nesmíme přesáhnout, často k tomu dochází při násobení proto jsou výsledky rovny modulu. Zde



Obr. 2.5: Ukázka promíchání sloupců za pomocí matic [22]

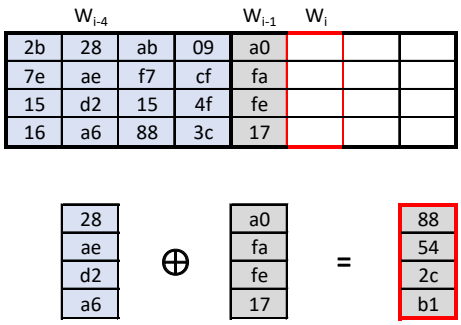
(GF(2^8)). Tedy hodnota po vynásobení může být 0 – 255 dekadicky nebo hexadecimálně 00 – FF.

Poslední krok je **vytvoření rundovního klíče**. Rundovní klíč je odvozen od vstupního a to tak, že vstupní klíč je rozdělen do matice 4×4 a zapisuje se po řádcích. Každá pozice v matici odpovídá jednomu Bytu. Vytvoření rundovního klíče poté probíhá následovně. Je vzat poslední sloupec matice a ten se cyklicky posune o jednu pozici směrem nahoru. Byty v něm jsou nahrazeny pomocí substituční tabulky (Tab. 2.9). Nově vzniklý sloupec se poté pomocí funkce XOR sečte s prvním sloupcem matice a zároveň i s prvním sloupcem matice Rcon. Tímto vznikne první sloupec matice rundovního klíče (Obr. 2.6). Další sloupec matice vzniká tak, že se spolu sčítá pomocí operace XOR nově vzniklý první sloupec matice a druhý sloupec matice původního klíče (Obr. 2.7). Třetí sloupec vzniká obdobně jako předchozí, tedy pomocí operace XOR se sčítá druhý sloupec nové matice a třetí sloupec původní matice (Obr. 2.8). Čtvrtý sloupec nové matice vzniká stejně jako předchozí dva, tedy znovu sčítáme za pomoci operace XOR třetí sloupec nové matice a čtrtý sloupec původní matice beze změny (Obr. 2.9). Tímto vznikl první rundovní klíč pro inicializační rundu. Postup pro vznik druhého rundovního klíče je naprosto shodný až na to, že se rotuje poslední sloupec z předchozí matice, který se za pomoci operace XOR sčítá s druhým sloupcem matice Rcon a prvním sloupcem předchozí matice, který je cyklicky posunutý a substituovaný (Obr. 2.10). Tento proces se opakuje pro všech 10 rundovních klíčů potřebných pro AES 128.

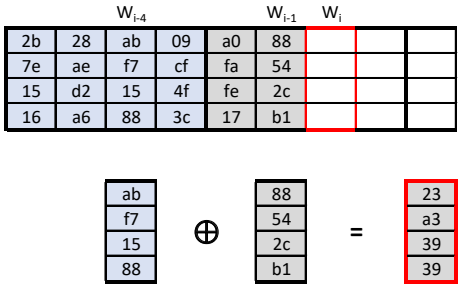


Obr. 2.6: Vytvoření prvního sloupce prvního rundovního klíče [33]

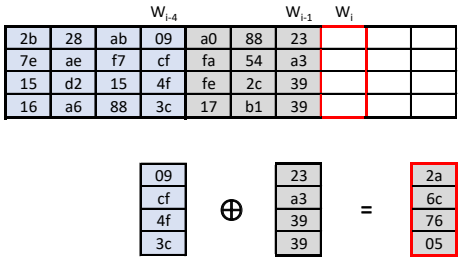
je modulo zadáno polynomem $x^8 + x^4 + x^3 + x + 1$, ten platí pro AES.



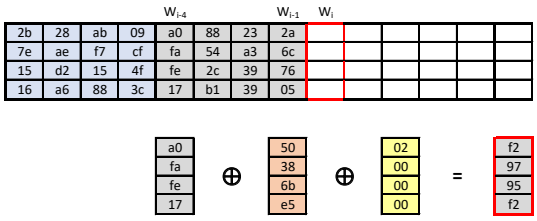
Obr. 2.7: Vytvoření druhého sloupce prvního rundovního klíče [33]



Obr. 2.8: Vytvoření třetího sloupce prvního rundovního klíče [33]



Obr. 2.9: Vytvoření čtvrtého sloupce prvního rundovního klíče [33]



Obr. 2.10: Vytvoření prvního sloupce druhého rundovního klíče [33]

V matici 4×10 Rcon je zajímavé, že se počítá pouze první řádek (Tab. 2.10). Zbylé řádky jsou nulové. Pro vytvoření prvního řádku pro jednotlivé sloupce platí:

- V případě, že se jedná o první sloupec je hodnota pozice $RC_1 = 1$

- Pro další sloupce je hodnota RC_i rovna

$$RC_i = 2 \cdot RC_{i-1} \tag{2.2}$$

  za podmínek $i > 1$ a hodnota předchozího $RC_{i-1} < 80_{16}$.
- Ale pokud jsou hodnoty $RC_{i-1} \geq 80_{16}$, pak se počítá jako

$$RC_i = 2 \cdot RC_{i-1} \oplus 11B_{16} \tag{2.3}$$

Výsledná matice poté vypadá následovně:

Tab. 2.10: Výsledná matice Rcon [33]

01	02	04	08	10	20	40	80	1b	36
00	00	00	00	00	00	00	00	00	00
00	00	00	00	00	00	00	00	00	00
00	00	00	00	00	00	00	00	00	00

Nyní, když jsou popsány všechny významné bloky funkce AES, je možné vysvětlit, jak funguje jako celek. V inicializační rundě je vstupující text sečtený pomocí matematické operace XOR s prvním rundovním klíčem. Výstup po tomto kroku pokračuje do hlavní rundy. Ta se opakuje 8 krát.

První proběhne **Nahrazení Bytů**, pak se pokračuje k posunu řádků. Po proběhnutí funkce dojde k pomíchání sloupců. Výsledek promíchání sloupců je sečten pomocí funkce XOR s druhým rundovním klíčem. Po proběhnutí hlavní rundy výsledek pokračuje do finální rundy. Ta je shodná s hlavní rundou, jen je vypuštěn krok promíchání sloupců. Výstupem je zašifrovaný text.

Dešifrování potom probíhá za pomoci inverzních funkcí, které se používají u šifrování, postup je naznačen na obrázku 2.3.

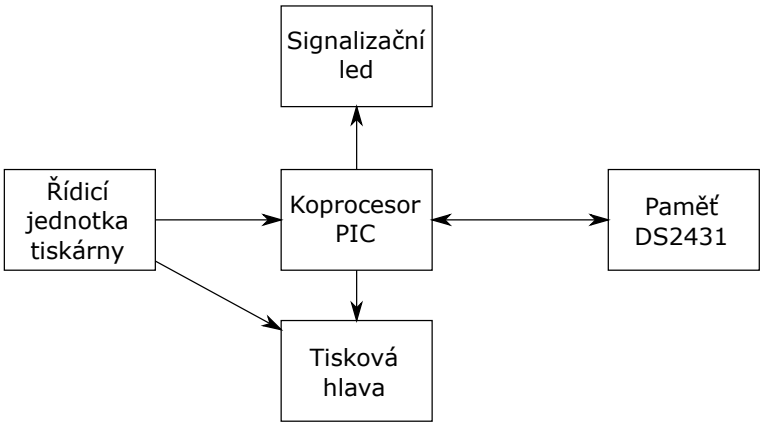
Šifrovací algoritmus AES byl vybrán z toho důvodu, že patří mezi dnes běžně používaný standard pro symetrické šifrování a překonal již DES i 3DES. Je rychlý a dá se nasadit i na mikrokontrolérech, které mají tak vysoký výkon jako běžné počítače. Dále je jeho výhodou, že potřebné matice nemusejí být uloženy přímo v paměti mikrokontroléru, která není příliš velká, ale není problém je vypočítat v každém kroku, protože se jedná o jednoduché operace.

3 Praktický návrh

Hlavním kritériem celého systému bylo vyvinout systém identifikace cartridge, který bude pracovat bez vnějšího zásahu uživatele. Celý systém musí být kompatibilní se stávající tiskárnou TSjet od firmy TS Electronics Zlín, s.r.o. a také snadno servisovatelný přímo u zákazníka nebo samotným zákazníkem v případě problémů.

3.1 Princip fungování systému zabezpečení

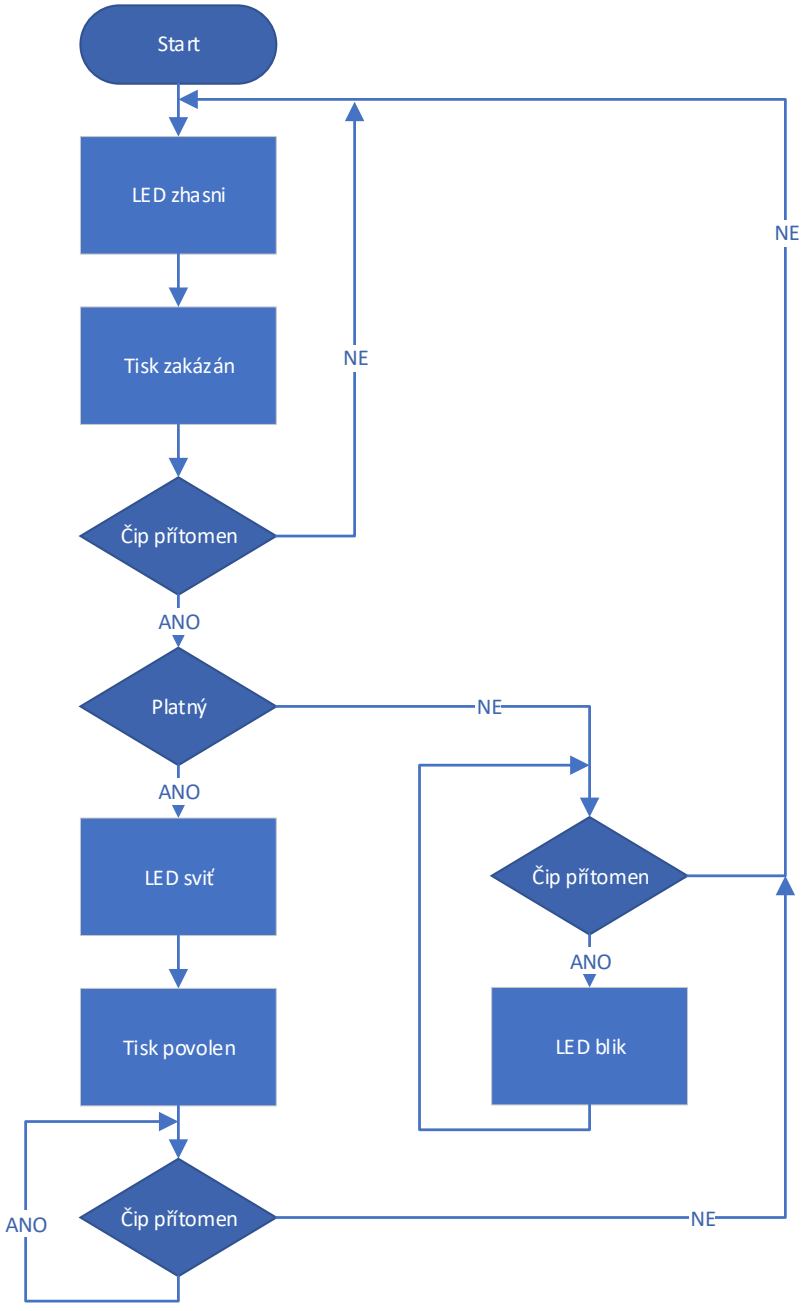
Při vytváření vlastního návrhu fungování systému se muselo dbát i na to, že nově navržený systém zabezpečení musí být zpětně kompatibilní se současnou tiskovou hlavou a řídicí jednotkou tiskárny. Současná tisková hlava tiskárny není schopna zpětně komunikovat s řídicí jednotkou, proto není možné zajistit ověření cartridge přímo hlavním procesorem tiskárny. Dále je tento procesor již využíván na 100 % své výpočetní kapacity, a není tedy schopen obstarávat ještě i verifikaci inkoustové cartridge, i kdyby byla zajištěna zpětná komunikace s+tiskovou hlavou. Proto bylo rozhodnuto o použití koprocesoru. Bude umístěn přímo do tiskové hlavy, aby nebylo potřeba řešit navíc nové propojení tiskové hlavy s+tiskárnou a tím bude zajištěna i zpětná kompatibilita systému. Když propojení a konektory zůstanou naprosto stejné a v případě závady je potřeba jen vyměnit tiskovou hlavu. Výměna tiskové hlavy není nijak složitá a zvládne ji provést i sám zákazník. Hlavní myšlenka fungování systému je zachycena na schématu 3.1.



Obr. 3.1: Schéma zabezpečení

Zabezpečení bude probíhat následujícím způsobem. V paměti je uložena předem spočítaná hodnota za pomoci zabezpečovacího algoritmu AES. Jako vstup do funkce AES (viz kapitola 2.2) slouží unikátní sériové číslo každého paměťového čipu. Po vložení cartridge do tiskové hlavy dojde k porovnání, zda se zašifrovaná hodnota

shoduje s hodnotou sériového čísla. Pokud dojde ke shodě, je povolen tisk, rozsvítí se signalizační dioda a trvale svítí. V případě, že je autorizace náplně neúspěšná, začne dioda blikat a tisk není povolen. V současné době není možné komunikovat s řídicí jednotkou obousměrně, řídicí jednotka komunikuje pouze směrem do tiskové hlavy, z toho důvodu je zde signalizační dioda. Průběh programu ověření cartridge je zachycen na vývojovém diagramu (Obr. 3.2).



Obr. 3.2: Vývojový diagram programu ověření cartridge

Obr. 3.3: Zapojení kompletní tiskárny TSjet [34]

3.2 Umístění a volba čipu

Jak již bylo popsáno v teoretické části práce (kapitola 1), řada výrobců opatřuje svoje inkoustové náplně identifikačním čipem. Je potřeba brát v potaz možnosti, kam se dá čip umístit tak, aby přidání tohoto bezpečnostního prvku nijak neomezilo zákazníka. Cartridge je vkládána do tiskové hlavy tak, že její přední strana dosedne na kontaktní pole pinů, které řídí trysky v inkoustové cartridgi, proto se může identifikační čip umístit na přední stranu cartridge a v držáku budou jen přidány piny pro jeho připojení. Dále je potřeba dbát na to, aby jeho umístění neomezovalo stávající vývody na přední straně.

Dalším nezbytným krokem je zvolení samotného čipu. Musí být vybírán s ohledem na prostor na přední straně cartridge, viz kapitola 1.1. Protože řada výrobců používá EEPROM paměť DS2431 (kapitola 1), byly prozkoumány její vlastnosti. Hlavní kritéria na to, aby mohla být použita, byla: fyzické rozměry pouzdra, ve kterém se vyrábí, možnost připojit se pouze kontaktními piny a počet zápisů. Na čelní straně cartridge není mnoho místa, které se dá využít k umístnění paměťového čipu, protože značnou část zabírá kontaktní pole pro řízení trysek hlavy a je zde také původní čip Hewlett-Packard. Co se týče fyzických rozměrů, nejvhodnější variantou je paměť DS2431 od Maxim Integrated™. Společnost čip vyrábí v pouzdře SFN s rozměry 3,5 mm×6,5 mm×0,75 mm. Dá se tedy bez problémů umístit na čelní stranu. Splňuje podmínku připojení kontaktními piny, protože plochy kontaktů na pouzdře jsou dostatečně velké, aby byl zajištěn bezproblémový spoj jen tlakem a nedocházelo k nechtěnému zkratování čipu při manipulaci s cartridgí.

Výrobce uvádí v technickém listě, že by měla vydržet 200 000 zápisů. [17] Předpokládaná životnost paměti je mnohonásobně větší než u cartridge. Čip je tedy ideální pro tyto účely, protože vyhovuje všem požadovaným kritériím. Paměťový čip od tohoto výrobce podporuje pouze komunikaci po 1-Wire®.

Z důvodů časové náročnosti pro použitý komunikační protokol, který se zde používá, se rozhodlo o použití koprocesoru, který bude zároveň zajišťovati i ověření pravosti náplně, aby se tímto krokem nezatěžoval hlavní procesor.

3.3 Výběr koprocesoru

Požadavky na koprocesor jsou dány především tím, že má být umístěn přímo v držáku cartridge. Musí tedy být k jeho plné funkčnosti použito co nejmenšího počtu součástek a musí podporovat vysoké přenosové rychlosti na komunikačních periferiích.

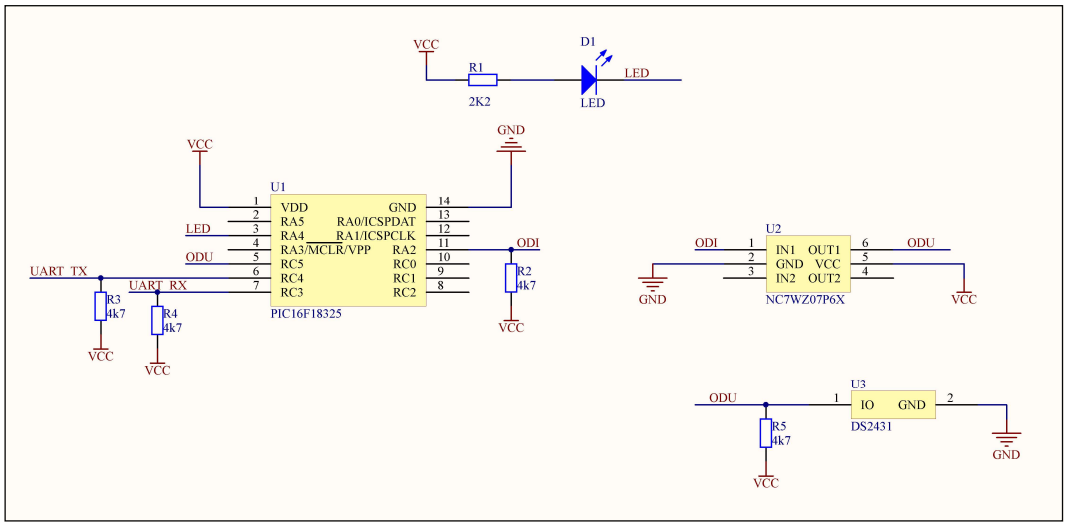
Jako hlavní procesor jednotky je použit mikrokontrolér založený na architektuře Intel 8051. Byl populární v době vzniku řídicí jednotky tiskárny, ale jeho nevýhodou je, že frekvence oscilátoru, která řídí kromě chodu vlastního procesoru i komunikační periferie, může být maximálně 22 MHz. Navíc musí být použit externí krystal pro tuto frekvenci. Z důvodu nutnosti externího krystalu pro tvorbu hodinového signálu a ne příliš jednoduchého programovaní bylo použití procesoru tohoto typu zamítnuto.

Protože aktuálně se ve firmě TS Electronics Zlín, s.r.o., která vyrábí průmyslové tiskárny, používají mikrokontroléry řady PIC od Microchip Technology, byl dán požadavek prodejci na doporučení konkrétního mikrokontroléru, který by vyhověl požadované aplikaci. Byl doporučen PIC16F18326. Jeho velkou výhodou je, že má již přímo ve svém pouzdře integrovaný krystal, který generuje frekvenci 32 MHz. Ta je využívána jak pro chod procesoru, tak pro řízení komunikačních periferií, proto není potřeba přidávat žádný další externí prvek. Mikrontrolér stačí připojit na napájecí napětí a může být provozován. Dále také podporuje takzvané aliasy pro jednotlivé piny portu. To znamená, že může být pin v deklaraci programu pojmenován a při psaní programu je používán už jen daný alias.

3.4 Konkrétní zapojení prototypového řešení

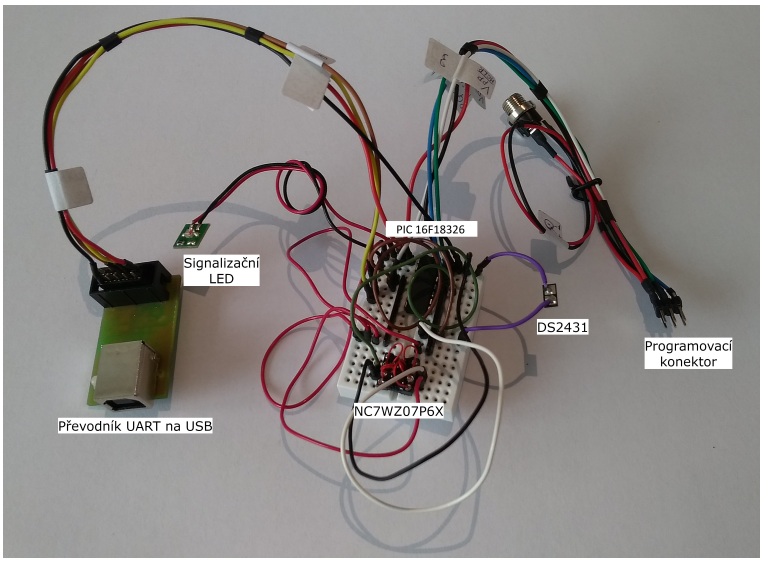
Na schématu 3.4 je zobrazeno konkrétní zapojení prototypového řešení.

Fyzické zapojení je provedeno v nepájivém poli. Toto řešení má velkou výhodu v tom, že se dá snad no modifikovat v případě potřeby.



Obr. 3.4: Schéma zapojení prototypu

V poli je umístěn samotný mikrokontrolér PIC16F18326. Pin RC4 je připojen na RX UART převodníku a pin RC3 je připojen na TX. Dále se v poli nachází otevřený drain (NC7WZ07P6X). Ten je napájený na patici, protože se nevyrábí v jiném pouzdře než pro montáž SMD. Na jeho vstup je připojen pin RA2, zároveň na jeho výstup je připojen pin RC5 a pin IO EEPROM paměti. Dále je k mikrokontroléru připojena na pinu RA4 signalizační led. V nepájivém poli se také nacházejí dvě odporová pole, která slouží jako pullup.



Obr. 3.5: Fotografie prototypového zapojení

3.5 Implementace komunikačního rozhraní

Pro komunikaci s pamětí se využívá sběrnice 1-Wire®. Pokud zařízení nepodporuje přímo komunikaci po této sběrnici, může se využít rozhraní UART. Toto rozhraní se na mikrokontroléru řídí programem. Nastavením odpovídajících registrů můžeme určit parametry tohoto rozhraní podle své potřeby. Názvy registrů a jejich rozložení se liší podle typu a značky použitého mikrokontroléru.

Jsou zde ukázky nejdůležitějších funkcí pro komunikaci pomocí UART po 1-Wire® sběrnici. Mezi ně patří funkce **Reset_PamPrit** viz 3.1. Funkce má za úkol zahájit komunikaci a ověřit, zda je paměť přítomná. Vrací logickou hodnotu. V případě, že je paměť přítomna, vrací se hodnota **TRUE**, v opačném případě je vracená hodnota **FALSE**.

Výpis 3.1: Funkce Reset_PamPrit

```
1  bool Reset_PamPrit() {
2      UartPam(207);
3      UartTx(0xF0);
4      __delay_ms(1);
5      if ((0x80 < RC1REG) && (RC1REG < 0xF0)) {
6          return true;
7      } else {
8          return false;
9      }
10 }
```

V Reset_PamPrit jsou použity další dvě funkce. A to **UartPam**, která nastavuje rozhraní UART na správné piny mikrokontroléru a modulační rychlost komunikace. Další je UartTx zajišťující vyslání odpovídající hodnoty pomocí UART, která do ní vstupuje jako parametr. Konkrétní podoba těchto dvou funkcí je závislá na použitém jednočipovém počítači, proto zde nejsou zobrazeny.

Další podstatnou částí kódu je funkce pro vyslání bajtu do paměti pomocí sběrnice 1-Wire® (Výpis 3.2).

Výpis 3.2: Funkce TxPamBajt

```
1  void TxPamBajt(char bajt) {
2      char nic;
3      char i;
4      for (i = 0; i < 8; i++) {
5          if (bajt & 1) {
6              TX1REG = 0xFF;
7          } else {
8              TX1REG = 0x00;
9          }
10         while (!TRMT);
11         bajt = bajt >> 1;
12         nic = RC1REG;
13         CREN = 0;
14         CREN = 1;
15     }
16 }
```

Jako parametr zde vstupuje hodnota Bytu, který má být vyslán po sběrnici. Je zde vidět, že pro vyslání 8 bitů na sběrnici musí rozhraní UART vyslat 8 Bytů. To je největší nevýhodou této sběrnice.

Obdobně to je i při přijímání Bytu. Zde zase na opak musí přijmout 8 Bytů, aby bylo možní přečíst jeden Byte. Paměť DS2431 vysílá jako první nejméně významný bit. K příjmu slouží funkce **RxPamBajt** (Výpis 3.3),která vrací hodnotu typu char.

Výpis 3.3: Funkce RxPamBajt

```
char RxPamBajt() {
    char pomP = 0x00;
    char vypn = 0xff;
    CREN = 0;
    CREN = 1;
    char i;
    for (i = 0; i < 8; i++) {
        if (vypn & 1) {
            TX1REG = 0xFF;
        } else {
            TX1REG = 0x00;
        }
        while (!TRMT);
        vypn = vypn >> 1;
        if (RC1REG == 0xFF) {
            pomP += 0b10000000;
        }
        CREN = 0;
        CREN = 1;
        if (i < 7) {
            pomP = pomP >> 1;
        }
    }
    return pomP;
}
```

Samotná paměť DS2431 se řídí sadou příkazů, které jsou popsány v technické dokumentaci výrobce [17].

3.6 Implementace zabezpečovacího algoritmu dat v paměti DS2431

Implementace šifrovacího algoritmu AES vychází z [35]. Jsou zde rozebrány jen ty části implementace, které neodpovídají doporučení výrobce mikrokontroléru, protože nebyly v tomto případě funkční.

Konkrétně je to funkce **inv_mix_column_optimization**, která se stará o zpětný převod míchání sloupců. V doporučení výrobce nefungoval převod posledního pole ve sloupci, proto byla upravena, tak aby dokázala korektně vracet původní hodnoty

před aplikováním funkce míchání sloupců. Nese název **DecMixColumn** (Výpis 3.4).

Výpis 3.4: Funkce DecMixColumn

```
void DecMixColumn(void) {
    char temp0;
    char temp1;
    char temp2;
    char temp3;
    char temp_0;
    char temp_1;
    char temp_2;
    char temp_3;
    char i;
    char j = 0;
    for (i = 0; i < 4; i++) {
        temp0 = block[j];
        temp1 = block[j + 1];
        temp2 = block[j + 2];
        temp3 = block[j + 3];
        temp_0 = temp0^temp1^temp2^temp3;
        temp_1 = Xtime(temp0^temp2);
        temp_2 = Xtime(temp1^temp3);
        temp_3 = Xtime(Xtime(temp_1^temp_2))^temp_0;
        block[j] = temp0^Xtime(temp0^temp1^temp_1)^temp_3;
        block[j + 1] = temp1^Xtime(temp1^temp2^temp_2)^temp_3;
        block[j + 2] = temp2^Xtime(temp2^temp3^temp_1)^temp_3;
        block[j + 3] = temp3^Xtime(temp0^temp3^temp_2)^temp_3;
        j = j + 4;
    }
}
```

Dále má výrobce ve svém doporučení jen teoreticky popsanou funkci posun řádku. Výsledný kód pro tuto funkci je zpracován následovně (Výpis 3.5).

Výpis 3.5: Funkce EncShiftRow

```
void EncShiftRow() {
    char i;
    char j;
    char temp;
    for (i = 1; i < 4; i++) {
        j = i;
        while (j--) {
            temp = block[i];
            block[i] = block[i + 4];
            block[i + 4] = block[i + 8];
            block[i + 8] = block[i + 12];
            block[i + 12] = temp;
        }
    }
}
```

Je zde použita i funkce **DecShiftRow**, ta má jen inverzní funkci, kdy vrací posunuté řádky zpět na původní pozici.

Obdobně teoreticky je v doporučení popsána i funkce pro generování klíčů pro šifrovaní a dešifrovaní. Funkce je zobrazena na výpisu 3.6 .

Výpis 3.6: Funkce EncKey

```
void EncKey() {
    char i;
    for (i = 0; i < 16; i++) {
        if (i < 4) {
            if (i == 3) {
                w[i] = w[i] ^SByt[w[i + 9]];
                w[0] = w[0]^rcon;
                rcon = Xtime(rcon);
            } else {
                w[i] = w[i] ^SByt[w[i + 13]];
            }
        } else {
            w[i] = w[i] ^ w[i - 4];
        }
    }
}
```

Výpis 3.7: Funkce DecKey

```
1  void DecKey() {
2      char i;
3      char j;
4      for (i = 0; i < 16; i++) {
5          j = 15 - i;
6          if (j < 4) {
7              if (j == 3) {
8                  w[j] = w[j] ^SByt[w[j + 9]];
9              } else {
10                 w[j] = w[j] ^SByt[w[j + 13]];
11             }
12             if (j == 0) {
13                 w[0] = w[0]^rcon;
14                 if (rcon & 0x01) {
15                     rcon = 0x80;
16                 } else {
17                     rcon = rcon >> 1;
18                 }
19             }
20         } else {
21             w[j] = w[j] ^ w[j - 4];
22         }
23     }
24 }
```

Pro správné fungování šifrování a dešifrování AES je potřeba mít definované následující globální konstanty a proměnné (Výpis 3.8).

Výpis 3.8: Definované globální konstanty a proměnné

```
1  const char SByt[256] = {
2  .
3  .
4  .};
5  const char RSByt[256] = {
6  .
7  .
8  .};
9  const char key[16] = {'P', 'O', 'K', 'U', 'S', 'P', 'O', 'K',
10 'U', 'S', 'P', 'O', 'K', 'U', 'S', '1'};
11 char kod[16];
12 char block[16];
13 char w[16];
14 char rcon = 0x01;
15 char round;
```

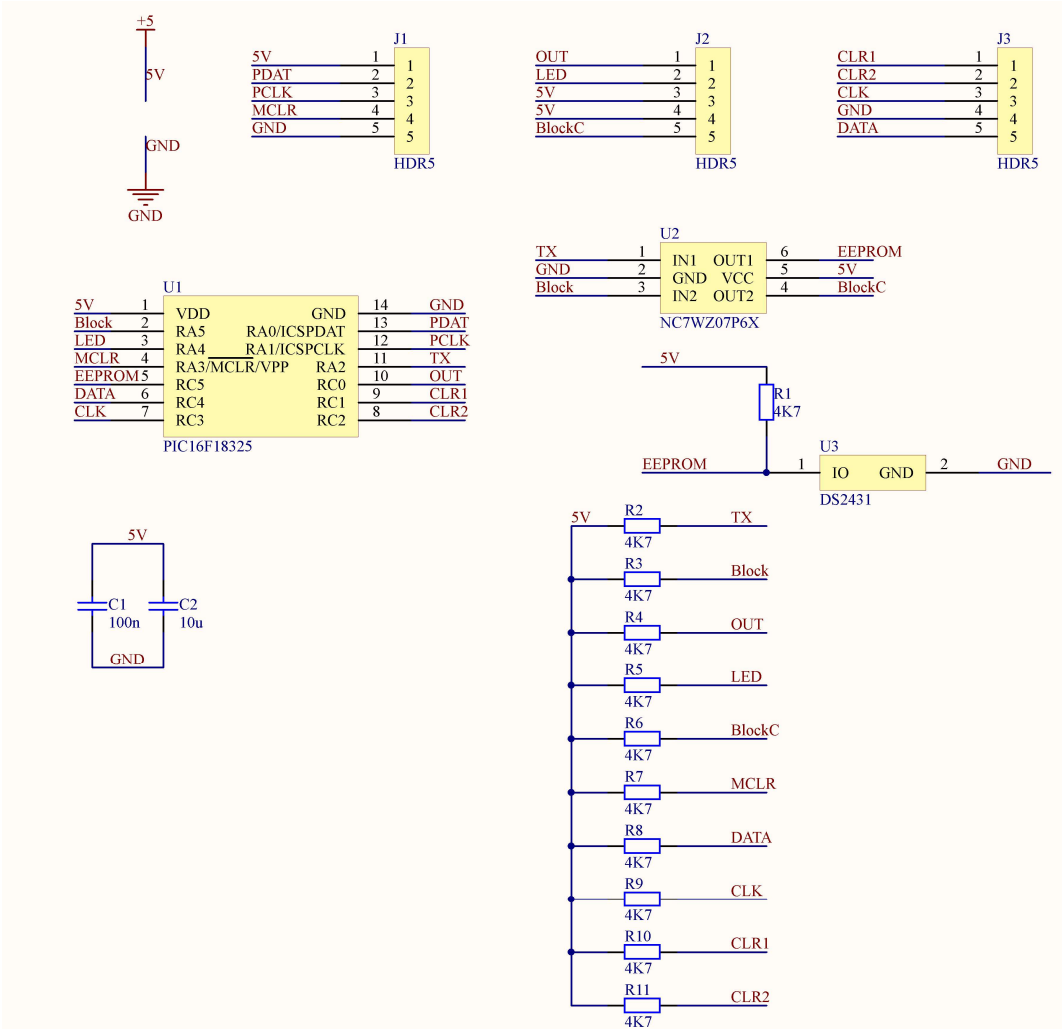
3.7 Návrh desky plošného spoje

Výsledná deska plošného spoje je navržena podle schématu 3.6.

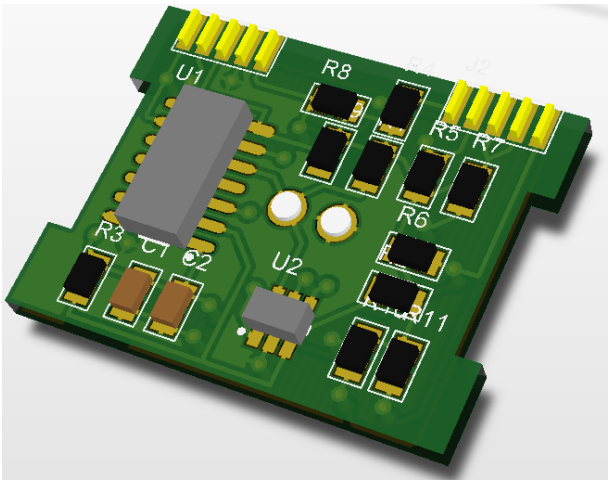
Rozměry desky jsou zvoleny dle požadavku firmy TS electronics Zlín, s.r.o., aby byla zajištěna kompatibilita s jejich produkty.

Použité součástky:

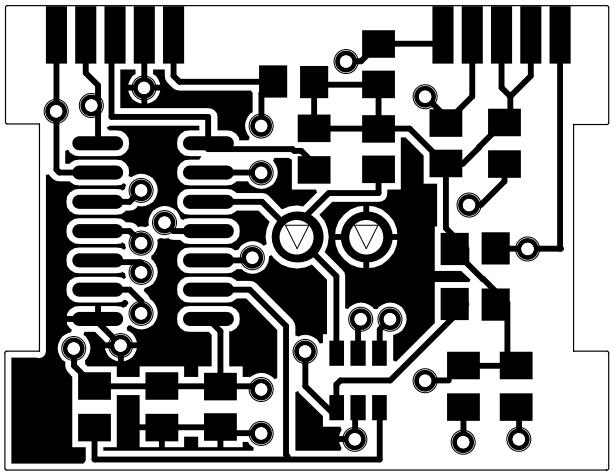
- R1 až R11: 4K7Ω
- C1: 100nF
- C2: 10μF
- U1: PC16F18326
- U2: NC7WZ07P6X
- U3: DS2431



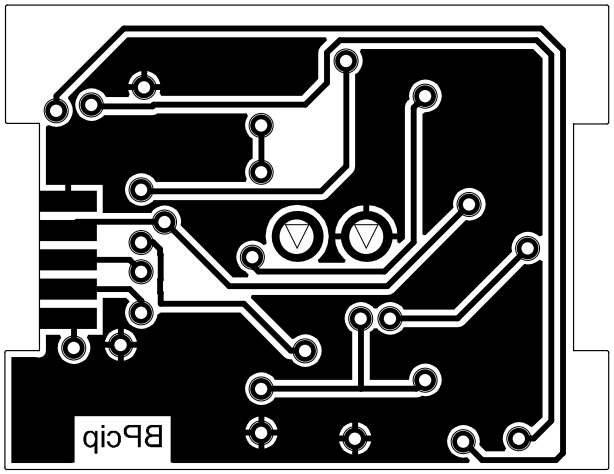
Obr. 3.6: Schéma zapojení výsledné desky plošného spoje



Obr. 3.7: Model výsledné desky plošného spoje



Obr. 3.8: Horní strana navržené desky plošného spoje



Obr. 3.9: Spodní strana navržené desky plošného spoje

3.8 Testování navrženého systému

Ověření systému proběhlo následujícím způsobem - Byla sestavena tiskárna TSjet s upravenou tiskovou hlavou, která obsahuje navrženou desku plošného spoje. Dalším krokem bylo osazení identifikačního čipu s nahranými daty na cartridge. Tímto způsobem bylo osazeno 10 kusů cartridgí. Čip se na cartridge upevňuje pomocí dvousložkového lepidla, které se vyznačuje velkou pevností po zaschnutí. Poté byla modifikovaná cartridge vložena do tiskové hlavy. Jakmile došlo k ověření a povolení tisku, byla poslaná tisková úloha úspěšně vytištěna. Postup testování byl pak ještě několikrát opakován. Během testovaní byly opakovaně rozpoznány všechny cartridge až na jednu. Zde byl zjištěn problém ve špatném umístění identifikačního čipu, který se během nalepovaní posunul. Přesné umístění čipu je kritickým bodem celého systému zabezpečení.

Druhým krokem testování bylo ověření zda je možné zjistit typ čipu pouhým odlepením z cartridge a přečtením kódového označení. Při všech pokusech došlo buď k mechanickému zničení čipu jeho rozlomením nebo byl po odstranění čipu z cartridge nápis s označením nečitelný. Proto by nemělo být jednoduché pouhým odlepení čipu zjistit jeho typ.

3.9 Shrnutí výsledků praktického návrhu

Otestování v praxi potvrdilo, že navržený systém je plně funkční i v průmyslovém prostředí a jeho bezpečnost plně dostačuje těmto účelům. Dle webu BlueKrypt je délka klíče 128 bit u symetrických šifer bezpečná do roku 2028 [36]. Jako další bezpečnostní opatření se na čip ukládají i další data. Do budoucna firma plánuje obousměrnou komunikaci tiskové hlavy s řídicí jednotkou, takže se předpokládá, že uložených dat na čipu ještě přibude.

Pokud by se někdo pokusil o prolomení tohoto zabezpečení, musel by vyřešit několik problémů. Jedním z nich je neznámý typ čipu. I kdyby by se mu z něj podařilo vyčíst data, musel by přesně vědět kde se nachází zašifrovaný obsah, který slouží k ověření. Pokud by se mu úspěšně podařilo tento obsah nalézt, musel by zjistit jakou šifrou je zabezpečený. V případě, že by odhalil typ použité šifry, musel by se ji pokusit prolomit takzvaně hrubou silou, tedy zkoušet všechny možnosti, které mohou nastat, což je časově velmi náročné při použití běžně dostupné výpočetní techniky. Jde řádově o roky až desítky let. Pokud klíč neunikne z firmy, jeho získání z prodávaných zařízení je nemožné, protože mikrokontrolér má nastavené příslušné registry tak, aby vyčtení uloženého programu z něj bylo nemožné.

Jelikož se tiskárna TSjet skládá z řídicí jednotky a tiskové hlavy, jsou tyto komponenty snadno nahraditelné v případě poruchy. Pokud by byla schopná tisková hlava, která obsahuje koprocesor, a jednotka schopny obousměrně komunikovat, bylo by možné, aby se tisková hlava autorizovala řídicí jednotce na její požádání. To by mohlo probíhat obdobným způsobem, jako probíhá autorizace cartridge, to znamená, že by hlavní procesor požádal koprocesor o to, aby se prokázal tak, že mu pošle zašifrovaná data. Ten by je následně rozklíčoval a zpět poslal dešifrovanou hodnotu nebo jen vrátil logickou hodnotu po porovnání s uloženými daty. U firmwaru by musela autorizace probíhat v hlavním procesoru jednotky, což v tomto případě znamená změnu celého návrhu elektroniky v řídicí jednotce, protože aktuální návrh a použité komponenty jsou využívány již na 100 % své využitelnosti. Z důvodů chybějící obousměrné komunikace tiskové hlavy a řídicí jednoty a také toho, že aktuální návrh elektroniky řídicí jednotky je využit již na své maximum, není možné snadno zajistit autorizaci jak hardwaru, tak i softwaru bez většího zásahu do stávajícího systému.

4 Závěr

V bakalářské práci jsou popsány základní možnosti identifikace inkoustové cartridge (viz kapitola 1). Jako nejvhodnější způsob u tiskárny TSjet se jeví použití identifikačního čipu.

Dále jsou porovnány v kapitole 2 jednotlivé zabezpečovací algoritmy pro uložená data v identifikačním čipu. Výběr algoritmu probíhal s ohledem na malý výpočetní výkon mikrokontroléru. Pro výslednou aplikaci byla zvolena symetrická šifra AES s délkou klíče 128 bit, která zajišťuje dostatečnou bezpečnost uložených dat na identifikačním čipu (viz kapitola3.9).

Bylo sestrojeno prototypové zapojení celého systému (viz kapitola 3.4), na kterém byly otestovány jeho základní vlastnosti. Pro komunikaci pomocí sběrnice 1-Wire® (viz 1.4) se používají příkazy pro ovládaní paměti DS2431 dle technické dokumentace výrobce [17]. Vedle komunikace byla testována i šifra AES, u níž byly zjištěny nesrovnalosti v doporučení výrobce [35], konkrétně nefunkčnost některých částí šifrovacího algoritmu. Ty byly posléze opraveny a uvedeny do funkčního stavu 3.6.

Navržený systém jako celek, skládající se z prototypového zapojení hardwaru a softwaru, bez problémů dokázal navázat komunikaci s čipem a ověřit, zda se jedná o čip zapsaný výrobcem a nebo někým jiným.

Jako poslední byla navržena a vyrobena deska plošného spoje s ohledem na kompatibilitu se současnou tiskovou hlavou tiskárny TSjet . Posléze byla do tiskové hlavy začleněna a celý systém tiskárny TSjet byl otestován. Z testování vyplynulo, že velmi kritické je přesné umístění samotného čipu na přední straně cartridge. Z tohoto důvodu bylo firmě TS Electronics Zlín, s.r.o. doporučeno v případě nasazení tohoto systému do praxe sestrojit šablonu pro nalepovaní těchto čipů, která by zajistila přesnou pozici každého čipu.

Literatura

[1] DOBŘICHOVSKÝ, Tomáš. Právní ochrana designu. In: *Enterprise Europe Network* [online]. Praha: Právní fakulta Univerzity Karlovy, 2011 [cit. 2018-11-24]. Dostupné z: <https://www.enterprise-europe-network.cz/files/dokums_raw/dobrichovskyipr-prumysldesignu_1318839816.pdf>.

[2] *Lapatushka* [online]. Rusko: Lapatuska, c2002-2018 [cit. 2018-11-24]. Dostupné z: <http://lapatushka.com/eng/index.htm>.

[3] FANG, Mei. How Is Inkjet Printing Done?. In: *SCIENCE 2.0* [online]. SCIENCE 2.0, c2018, April 12th 2010 [cit. 2018-11-24]. Dostupné z: <https://www.science20.com/mei/how_inkjet_printing_done>.

[4] HP 45 Large Black Original Ink Cartridge. *HP Online Store* [online]. Palo Alto (CA): Hewlett-Packard, c2018 [cit. 2018-11-24]. Dostupné z: <https://store.hp.com/UKStore/Merch/Product.aspx?id=51645AE&opt=&sel=SUP>.

[5] MEDLA, Eduard. MODERNÍ ZPŮSOBY PROGRAMOVÁNÍ MIKROKONTROLÉRU. In: *VUT v Brně* [online]. Brno: VUT v Brně, 2015 [cit. 2018-11-24]. Dostupné z: <https://www.vutbr.cz/www_base/zav_prace_soubor_verejne.php?file_id=104784>.

[6] OLIVKA, Petr. Procesory CISC a RISC. In: *VŠB — Technická univerzita Ostrava* [online]. Ostrava: VŠB — Technická univerzita Ostrava, 2010 [cit. 2018-11-25]. Dostupné z: <http://poli.cs.vsb.cz/edu/arp/down/procrisc.pdf>.

[7] AT89 Series Hardware Description. In: *DOCPLAYER* [online]. San Jose (CA): Atmel, 1997, 12/97 [cit. 2018-11-25]. Dostupné z: <https://goo.gl/KpGYhn>.

[8] Intel 8051. In: *Wikipedia: the free encyclopedia* [online]. San Francisco (CA): Wikimedia Foundation, 2018 [cit. 2018-11-25]. Dostupné z: <https://cs.wikipedia.org/wiki/Intel_8051>.

[9] PREDKO, Michael. *Programming and customizing the PIC microcontroller*. 3rd ed. New York: McGraw Hill, c2008. ISBN 978-0-07-147287-6..

[10] PIC16F18325. *Microchip* [online]. Chandler (AZ): Microchip Technology, c1998-2018 [cit. 2018-11-25]. Dostupné z: <https://www.microchip.com/wwwproducts/en/PIC16F18325>.

[11] PIC16(L)F18325/18345. In: *Microchip: PIC16F18325* [online]. Chandler (AZ): Microchip Technology, c2015-2018 [cit. 2018-11-29]. Dostupné z:

<http://ww1.microchip.com/downloads/en/DeviceDoc/PIC16-L-F18325_18345-Data-Sheet-40001795G.pdf>.

[12] Sběrnice. In: *Wikipedia: the free encyclopedia* [online]. San Francisco (CA): Wikimedia Foundation, 2018 [cit. 2018-11-26]. Dostupné z: <https://cs.wikipedia.org/wiki/Sb%C4%9Brnice>.

[13] Sběrnice 1-Wire. In: *Vyvoj.hw.cz: profesionální elektronika* [online]. Praha: HW server, c1997-2014, 17. Listopad 2004 [cit. 2018-11-26]. Dostupné z: <https://vyvoj.hw.cz/navrh-obvodu/rozhrani/sbernice-1-wiretm.html>.

[14] Using a UART to Implement a 1-Wire Bus Master. *Maxim Integrated* [online]. San Jose (CA): Maxim Integrated, 2002, 10 Sep [cit. 2018-11-26]. Dostupné z: <https://www.maximintegrated.com/en/app-notes/index.mvp/id/214>.

[15] Historie a současnost datových úložišť. In: *SVĚT HARDWARE: ... vše ze světa počítačů* [online]. Brno: oXy Online, 2018 [cit. 2018-11-26]. Dostupné z: <https://www.svethardware.cz/historie-a-soucasnost-datovych-ulozist/23935>.

[16] Vnitřní paměti. In: *Masarikova univerzita: FAKULTA INFORMATIKY* [online]. Brno: Masarikova univerzita, 2018 [cit. 2018-11-26]. Dostupné z: <https://www.fi.muni.cz/usr/pelikan/ARCHIT/TEXTY/INTPAM.HTML>.

[17] DS2431: 1024-Bit, 1-Wire EEPROM. In: *Maxim Integrated* [online]. San Jose (CA): Maxim Integrated, c2018, 3/15 [cit. 2018-11-29]. Dostupné z: <https://datasheets.maximintegrated.com/en/ds/DS2431.pdf>.

[18] BEŠTA, Miloš. TRANZISTORY. In: *Studijní materiály elektro: pro učební obor elektrikář — slaboproud* [online]. Krupka: studijní materiály elektro, c2018 [cit. 2018-11-26]. Dostupné z: <http://www.mbest.cz/wp-content/uploads/2013/01/T1.5-Tranzistor.pdf>.

[19] This Month in Physics History: November 17 - December 23, 1947: Invention of the First Transistor. *APS NEWS* [online]. 2000, November 2000, **9**(10), 1 [cit. 2018-11-29]. Dostupné z: <https://www.aps.org/publications/apsnews/200011/history.cfm>.

[20] THORNTON, Scott. What is an open drain?. In: *MICROCONTROLLER TiPS* [online]. Cleveland (OH): WTWH Media, c2018, July 6, 2017 [cit. 2018-11-26]. Dostupné z: <https://www.microcontrollertips.com/what-is-an-open-drain-faq/>.

[21] HAJNÝ, Jan. Přednáška předmětu BZKR - Základy kryptografie 1. In: *VUT v Brně: Elearning* [online]. Brno: VUT v Brně, 2018 [cit. 2018-11-27]. Dostupné z: <https://moodle.vutbr.cz/mod/resource/view.php?id=218674>.

[22] HAJNÝ, Jan. Přednáška předmětu BZKR - Základy kryptografie 8. In: *VUT v Brně: Elearning* [online]. Brno: VUT v Brně, 2018 [cit. 2018-11-27]. Dostupné z: <https://moodle.vutbr.cz/mod/resource/view.php?id=231701>.

[23] A Detailed Description of DES and 3DES Algorithms (Data Encryption Standard and Triple DES). In: *Commonlounge* [online]. Commonlounge, 2018 [cit. 2018-11-27]. Dostupné z: <https://www.commonlounge.com/discussion/5c7c2828bf6b4724b806a9013a5a4b99>.

[24] SPEKKING, Raimond. HP Black Print Cartridge 45-5062. In: *Wikipedia: the free encyclopedia* [online]. San Francisco (CA): Wikimedia Foundation, 2018, 24 November 2017 [cit. 2018-11-27]. Dostupné z: <https://commons.wikimedia.org/wiki/File:HP_Black_Print_Cartridge_45-5062.jpg>.

[25] ŠICHNÁREK, Pavel. *KARTRIDŽ V2 2016*. Zlín: TS Electronics Zlín, 2018.

[26] PLUGWASH. Puerto serie Rs232. In: *Wikipedia: the free encyclopedia* [online]. San Francisco (CA): Wikimedia Foundation, 2018, 2006-08-11 [cit. 2018-11-27]. Dostupné z: <https://commons.wikimedia.org/wiki/File:Puerto_serie_Rs232.png>.

[27] KOVÁŘ, Josef. *HW_i51*. Zlín, 2014.

[28] OMEGATRON. BJT NPN symbol. In: *Wikipedia: the free encyclopedia* [online]. San Francisco (CA): Wikimedia Foundation, 2018, 13 November 2007 [cit. 2018-11-28]. Dostupné z: <https://commons.wikimedia.org/wiki/File:BJT_NPN_symbol.svg>.

[29] OMEGATRON. BJT PNP symbol. In: *Wikipedia: the free encyclopedia* [online]. San Francisco (CA): Wikimedia Foundation, 2018, 13 November 2007 [cit. 2018-11-28]. Dostupné z: <https://commons.wikimedia.org/wiki/File:BJT_PNP_symbol.svg>.

[30] Unipolární tranzistor. In: *Wikipedia: the free encyclopedia* [online]. San Francisco (CA): Wikimedia Foundation, 2018 [cit. 2018-11-28]. Dostupné z: <https://cs.wikipedia.org/wiki/Unipol%C3%A1rn%C3%AD_tranzistor>.

[31] The Advanced Encryption Standard (AES) Algorithm. In: *Commonlounge* [online]. Commonlounge, 2018, leden [cit. 2018-11-30]. Dostupné z: <https://www.commonlounge.com/discussion/290ee9a2afcb40fdaa21f0a03285832f>.

[32] ADVANCED ENCRYPTION STANDARD (AES). In: *NIST* [online]. Gaithersburg (MD): NIST, 2001, November 26 [cit. 2018-12-01]. Dostupné z: <https://nvlpubs.nist.gov/nistpubs/FIPS/NIST.FIPS.197.pdf>.

[33] AES animation - Forma Estudio. In: *Forma Estudio* [online]. Montevideo (URY): Forma Estudio [cit. 2018-12-01]. Dostupné z: <http://www.formaestudio.com/rijndaelinspector/archivos/Rijndael_Animation_v4_eng.swf>.

[34] TSjet201. In: *TS Electronics Zlín, s.r.o.* [online]. Zlín: TS Electronics Zlín, c2011 [cit. 2018-12-02]. Dostupné z: <http://www.tsezlin.cz/cs/produkty/prumyslove-tiskarny/TSjet201.jpg>.

[35] AN821: Advanced Encryption Standard Using the PIC16XXX. In: *Microchip* [online]. Chandler (AZ): Microchip Technology, 2004, 14.03.2004 [cit. 2019-04-27]. Dostupné z: <http://ww1.microchip.com/downloads/en/AppNotes/00821a.pdf>.

[36] BlueKrypt: Cryptographic Key Length Recommendation. *BlueKrypt* [online]. Belgie: BlueKrypt, 2018, 10. 6. 2018 [cit. 2019-05-15]. Dostupné z: <https://www.keylength.com/en/3/>.

Seznam symbolů, veličin a zkratek

HP	Hewlett-Packard
MCU	jednočipový počítač – microcontroller unit
A/D	analogově digitální
b	bit
RISC	redukovaná instrukční sada – Reduced Instruction Set Computer
CISC	kompletní instrukční sada – Complex Instruction Set Computer
USB	universální sériová sběrnice – Universal Serial Bus
I^2C	multi-masterová počítačová sériová sběrnice – Inter-Integrated Circuit
SPI	sériové periferní rozhraní – Serial Peripheral Interface
CAN	Controller Area Network
UART	univerzální asynchronní sériové rozhraní – Universal Asynchronous Receiver and Transmitter
RAM	paměť s libovolným přístupem – Random Access Memory
ROM	paměť umožnující jen čtení – Read-Only Memory
SRAM	statická paměť s náhodným přístupem – Static Random Access Memory
EEPROM	elektronicky programovatelná paměť určená pouze pro čtení – Electrically Erasable Programmable Read-Only Memory
USART	univerzální synchronní / asynchronní sériové rozhraní – Universal Synchronous / Asynchronous Receiver and Transmitter
log.	logická hodnota úrovně signálu
TX	vysílaní dat – transmit data
RX	příjem dat – receive data
BR	modulační rychlost – Baud rate
FET	tranzistory řízené elektrickým polem – Field-Effect Transistors
debuger	nástroj na odladění programu
ml	mililitry
pl	pikolitry
kB	kilobajt
V	volt
pF	pikofarad
kb/s	kilobit za sekundu
μ**s**	mikrosekunda
MHz	megahertz
f_{vz}	vzorkovací kmitočet

Seznam příloh

A Obsah přiloženého CD

Programy BPcipCtecka.c, BPcipHlava.c a PBcipZapisovac.c byly vytvořeny v MPLAB X IDE v5.15 a kompilovány v XC8(v2.05).

- `/` ..... Kořenový adresář přiloženého CD
  - `PlosnySpoj` ..... Soubory plošneho spoje
    - `BPcip` ..... Hlavní adresář návrhu desky
      - `Project Outputs for BPcip`
        - `...`
      - `BPcip.CSPcbDoc`
      - `BPcip.PrjPcb`
      - `BPcip.PrjPcbStructure`
      - `BPcip.SchDoc`
  - `Programy` ..... Programy vytvoření v rámci BP
    - `BPcipCtecka.X` ..... Obsahuje program pro ověření zápisu na čip
      - `BPcipCtecka.c`
      - `...`
    - `BPcipHlava.X` ..... Obsahuje program pro ověření čipu v hlavě tiskárny
      - `BPcipHlava.c`
      - `...`
    - `BPcipZapisovac.X` ..... Obsahuje program pro zapsání dat na čip
      - `PBcipZapisovac.c`
      - `...`
  - `195300_Galad_BP_v5.pdf` ..... Hlavní dokument BP